



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV RADIOELEKTRONIKY

DEPARTMENT OF RADIO ELECTRONICS

VYSOKORYCHLOSTNÍ AKVIZIČNÍ SYSTÉM

HIGH SPEED ACQUISITION SYSTEM

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE

AUTHOR

Bc. Tomáš Svoboda

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Michal Kubíček, Ph.D.

BRNO 2018



Diplomová práce

magisterský navazující studijní obor **Elektronika a sdělovací technika**

Ústav radioelektroniky

Student: Bc. Tomáš Svoboda

ID: 164408

Ročník: 2

Akademický rok: 2017/18

NÁZEV TÉMATU:

Vysokorychlostní akviziční systém

POKyny PRO VYPRACOVÁNÍ:

Navrhnete sestavu měřicího řetězce vysokorychlostního analogového měření s rychlostí vzorkování minimálně 4 GS/s a rozlišením min. 8 bitů. Provedte výběr dostupných A/D převodníků, obvodů FPGA pro zpracování digitálního signálu a paměťových obvodů schopných průběžně ukládat výsledky měření o objemu minimálně milionu vzorků. Zvažte použití hotových komerčně dostupných modulů. Uveďte a porovnejte možné způsoby přenosu naměřených dat do počítače. Posuďte cenovou náročnost možných řešení, jejich výhody a nevýhody a stanovte konstrukční limity pro dosažitelnou rychlost vzorkování a rozlišení vzhledem k aktuální součástkové základně.

Sestavte a funkčně ověřte vybraný měřicí řetězec s použitím komerčně dostupných modulů, včetně exportu naměřených hodnot do počítače a jejich vyhodnocení.

DOPORUČENÁ LITERATURA:

[1] MAXFIELD, C. The Design Warrior's Guide to FPGAs. 1st ed. Newnes – Elsevier, Burlington, MA, 2004. 542 p. ISBN 0-7506-7604-3

[2] CATSOULIS, J. Designing embedded hardware. 2nd ed. Sebastopol: O'Reilly, 2005, xvi, 377 s. ISBN 05-960-0755-8.

Termín zadání: 5. 2. 2018

Termín odevzdání: 17. 5. 2018

Vedoucí práce: Ing. Michal Kubíček, Ph.D.

Konzultant: Ing. Jan Cupák



prof. Ing. Tomáš Kratochvíl, Ph.D.
předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

ABSTRAKT

Tato diplomová práce se zaměřuje na návrh vysokorychlostního akvizčního systému za použití obvodu FPGA a vysokorychlostního A/D převodníku s moderním rozhraním JESD204B. Vzhledem k požadované rychlosti vzorkování naráží práce na omezené možnosti dnešní součástkové základny. V práci je proto provedena studie trhu dnes dostupných obvodů a hotových modulů. Výsledný návrh je postaven na bázi dostupných vývojových modulů, pomocí kterých je dosaženo vzorkovací rychlosti až 5 GSa/s při 12bitovém rozlišení. Získaná data jsou zaslána do počítače přes rozhraní Ethernet za použití lwIP stacku a jádra TEMAC na procesoru Microblaze.

KLÍČOVÁ SLOVA

FPGA, AD převodník, ADC, Xilinx, JESD204B, Microblaze, lwIP

ABSTRACT

This master's thesis is focused on the design of a highspeed aqizition system which is based on FPGA and a highspeed AD converter with modern JESD204B interface. Considering the requirements, such as high samplig rate, the current range of available devices is limited. Therefore the market overview of the modern IC and modules was made. The resulting design is based on available modules, so the rached sampling rate is up to 5 GSa/s with 12bits resolution. Data from measurement are send to PC via Ethernet which uses lwIp stack and TEMAC core on Microblaze proccessor.

KEYWORDS

FPGA, AD converter, ADC, Xilinx, JESD204B, Microblaze, lwIP

SVOBODA, T. *Vysokorychlostní akviziční systém*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav radioelektroniky, 2018. 78 s., 2 s. příloh. Diplomová práce. Vedoucí práce: Ing. Michal Kubíček, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svoji diplomovou práci na téma Vysokorychlostní akviziční systém jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

(podpis autora)

PODĚKOVÁNÍ

Tímto bych rád poděkoval Ing. Michalovi Kubíčkovi, Ph.D. za pomoc se zpracováním diplomové práce, jeho kritické připomínky a v neposlední řadě za jeho čas a nápady při ladění JESD204B rozhraní. Poděkování patří také firmě OZM Research s.r.o a Ing. Janovi Cupákovi za možnost zpracování této práce.

Experimentální část této diplomové práce byla realizována na výzkumné infrastruktuře
vybudované v rámci projektu CZ.1.05/2.1.00/03.0072
Centrum senzorických, informačních a komunikačních systémů (SIX)
operačního programu Výzkum a vývoj pro inovace.

OBSAH

Seznam obrázků	x
Seznam tabulek	xii
Úvod	1
1 A/D převod	2
1.1 Průběh A/D převodu	2
1.2 Typy A/D převodníků	4
1.3 Vybrané parametry A/D převodníků	5
1.3.1 Efektivní počet bitů	5
1.3.2 Rychlost vzorkování a šířka pásma	6
1.3.3 Dynamický rozsah bez IM zkreslení	6
1.3.4 Další důležité parametry:	7
1.4 Zvyšování vzorkovací frekvence	8
2 Obvody FPGA	10
2.1 Popis	10
2.2 Vysokorychlostní transceivery	11
2.3 Konfigurace obvodů FPGA	13
3 Paměti	14
3.1 Parametry pamětí	14
3.1.1 Kapacita	14
3.1.2 Šířka sběrnice	14
3.1.3 Frekvence, rychlost	14
3.1.4 Šířka pásma	14
3.1.5 Časové parametry	15
3.1.6 Datová propustnost	15
3.2 Typy pamětí	15
3.2.1 DRAM vs SRAM	15
3.2.2 Paměti typu DDRX SDRAM	15
3.2.3 Paměti typu QDR	16

3.2.4	Paměti typu LPDDR	17
3.2.5	Paměti typu RDRAM	17
3.2.6	Paměti HBM a HMC	17
3.2.7	Porovnání	18
4	Standard JESD204	19
4.1	Úvod	19
4.2	Vrstvy rozhraní	20
4.2.1	Fyzická vrstva	20
4.2.2	Datová vrstva	22
4.2.3	Transportní vrstva	23
4.2.4	Aplikační vrstva	24
4.3	Skupinová synchronizace a inicializace	24
4.4	Porovnání s LVDS	25
5	Aplikace Ethernetu	26
5.1	Zapouzdření dat v síti TCP/IP	26
5.2	Protokoly v síti TCP/IP	26
5.2.1	Protokol TCP	27
5.2.2	Protokol UDP	28
5.3	Ethernet	28
6	Řešení HW části	30
6.1	Vstupní požadavky	30
6.2	Analýza trhu a výběr komponent	30
6.2.1	A/D převodníky	30
6.2.2	A/D moduly	31
6.2.3	Obvody FPGA	32
6.2.4	Vývojové desky s FPGA	33
6.2.5	Paměťové obvody	34
6.2.6	Paměťové moduly	34
6.2.7	Kompletní řešení	34
6.3	Popis hardwaru	35
6.3.1	Blokové schéma	35
6.3.2	Standard VITA 57.1 a 57.4	35
6.3.3	Vývojová deska s FPGA Genesys 2	36

6.3.4	ADC vývojová deska ADC12DJ3200EVM	37
7	FPGA design	40
7.1	Koncept implementace	40
7.2	Programování FPGA Xilinx	42
7.3	Konfigurace ADC	43
7.4	JESD204B rozhraní	46
7.4.1	Implementace fyzické vrstvy	46
7.4.2	Implementace linkové vrstvy	48
7.4.3	Implementace transportní vrstvy	51
7.5	Ukládání dat	52
7.6	Čtení a odesílání dat	55
8	Příjem dat	58
9	Testování JESD204B a ADC	61
9.1	Testovací sekvence A/D převodníku	61
9.1.1	Kontinuální zasílání K28.5 znaků	61
9.1.2	Kontinuální zasílání D21.5 znaků	61
9.1.3	Zasílání dat rampové funkce	62
9.1.4	Zasílání dat testovací transportní funkce	63
9.1.5	Opakované zasílání ILAS a analýza konfiguračních dat z ILAS	63
9.1.6	Diskuze	65
9.2	Testování na reálných signálech	67
	Závěr	71
	Literatura	72
	Seznam symbolů, veličin a zkratk	76
	Obsah elektronické přílohy	78
A	Fotodokumentace	79
B	Blokový design – Vivado	79

SEZNAM OBRÁZKŮ

Obrázek 1.1	Průběh vzorkování a kvantování při A/D převodu, převzato z [41]	2
Obrázek 1.2	Jednoduchý obvod typu Sample and hold	2
Obrázek 1.3	Znázornění Nyquistových zón pro dostatečnou vzorkovací frekvenci (A), pro malou vzorkovací frekvenci (B)	3
Obrázek 1.4	Využití A/D převodníků z hlediska rychlosti a rozlišení, převzato z [42]	5
Obrázek 1.5	Znázornění pojmu SFDR ve spektru, převzato z [5]	7
Obrázek 1.6	Princip prokládání při dvou ADC	8
Obrázek 2.1	Stavba logické buňky obvodu FPGA, převzato z [44]	10
Obrázek 2.2	Struktura obvodu FPGA, převzato z [44]	11
Obrázek 2.3	Vnitřní struktura transceiveru, převzato z [45]	12
Obrázek 3.1	Průběh přenosu dat u paměti QDR, převzato z [43]	16
Obrázek 3.2	Vnitřní struktura paměti typu HMC, převzato z [18]	17
Obrázek 3.3	Integrovaná paměť HBM na FPGA XCVU37P firmy Xilinx, převzato z [19]	18
Obrázek 4.1	Vrstvy a bloky rozhraní JESD204B	20
Obrázek 4.2	Fyzické provedení JESD204B	21
Obrázek 4.3	Základní struktury JESD204B	23
Obrázek 4.4	Hranice rámce a multirámce v datovém toku	23
Obrázek 4.5	Postup při CGS a ILAS	24
Obrázek 4.6	Inicializační sekvence v rozhraní JESD204B	25
Obrázek 5.1	Zapouzdření dat v síti TCP/IP	26
Obrázek 5.2	Odesílání paketů prostřednictvím protokolu TCP, metoda Stop and Wait	27
Obrázek 5.3	TCP segment	28
Obrázek 5.4	UDP datagram	28
Obrázek 5.5	Rámec Ethernetu	29
Obrázek 6.1	Blokové schéma navrhovaného systému	35
Obrázek 6.2	Konektor FMC + (vlevo) a FMC (vpravo), převzato z [48]	36
Obrázek 6.3	Vývojová deska Genesys 2	37
Obrázek 6.4	Vývojová deska ADC12DJ3200EVM	38
Obrázek 6.5	Vnitřní struktura A/D převodníku ADC12DJ3200	38

Obrázek 6.6	GUI pro konfigurace ADC (na obrázku nastavení výstupních hodin) ...	39
Obrázek 7.1	Implementace bloků do obvodu FPGA	40
Obrázek 7.2	Tvorba blokového designu v prostředí Vivado.....	42
Obrázek 7.3	Průběh SPI konfigurace, A – standardní mód, B – streaming mód, převzato z [46]	45
Obrázek 7.4	Příklad zachycených dat CGS analyzátozem pro špatně nastavenou ekvalizaci	48
Obrázek 7.5	Možnosti konfigurace JESD204B přijímače, použitá konfigurace	49
Obrázek 7.6	Dvouportová paměť BRAM, převzato z [47]	52
Obrázek 7.7	Použití BRAM ve spojení s BRAM kontrolérem	53
Obrázek 7.8	Základní princip činnosti bloku ukládání, simulace	54
Obrázek 7.9	Vybrané signály pro práci s pamětí, získané analyzátozem.....	54
Obrázek 7.10	Zachycené pakety programem Wireshark (data testu transportní vrstvy)	57
Obrázek 8.1	Příjem testovacích dat.....	59
Obrázek 8.2	Přijatá reálná data z ADC	60
Obrázek 9.1	Zachycený příjem znaků K28.5	61
Obrázek 9.2	Zachycený příjem znaků D21.5	62
Obrázek 9.3	Zachycená data rampové funkce pomocí analyzátoru (Chipscope)	62
Obrázek 9.4	Zachycená data pro mód testu transportní vrstvy	63
Obrázek 9.5	Zachycená ILAS pomocí analyzátoru (Chipscope)	63
Obrázek 9.6	Data fyzické vrstvy pro normální mód	66
Obrázek 9.7	Data fyzické vrstvy pro opakovanou ILA.....	66
Obrázek 9.8	Průběh signálu pro vstupní sinusový signál, $f = 2,5 \text{ GHz}$	69
Obrázek 9.9	Normované amplitudové spektrum pro sinusový vstupní signál,.....	69
Obrázek 9.10	Průběh signálu pro vstupní sinusový signál, $f = 200 \text{ MHz}$	70
Obrázek 9.11	Normované amplitudové spektrum pro sinusový vstupní signál,.....	70

SEZNAM TABULEK

Tabulka 2.1	Možností transceiverů FPGA obvodů firmy Xilinx [14].....	12
Tabulka 3.1	Porovnání typů paměti HBM, HMC, DDR4 a RLD RAM3 [19], [20] ...	18
Tabulka 4.1	Porovnání revizí standardu JESD204X v kontextu s LVDS [11].....	19
Tabulka 4.2	Možné speed-grady v rámci fyzické vrstvy JESD204B [23]	21
Tabulka 4.3	Řídící znaky využití rozhraní JESD204B [23]	22
Tabulka 4.4	Porovnání fyzické vrstvy rozhraní LVDS a JESD204B [25]	25
Tabulka 6.1	Přehled vyhovujících A/D převodníků pro požadované parametry.....	31
Tabulka 6.2	Přehled dostupných A/D modulů pro požadované parametry - výběr....	31
Tabulka 6.3	Přehled vyhovujících FPGA obvodů - výběr.....	32
Tabulka 6.4	Přehled vyhovujících vývojových desek firmy Xilinx - výběr.....	33
Tabulka 6.5	Vyhovující paměťové obvody - výběr.....	34
Tabulka 6.6	Základní vlastnosti FPGA XC7K325T [28]	36
Tabulka 6.7	Základní vlastnosti ADC ADC12DJ3200 [29].....	37
Tabulka 7.1	Nastavené parametry fyzické vrstvy.....	47
Tabulka 7.2	Možné módy přenosu A/D převodníky (příklady)	50
Tabulka 7.3	Mapování vzorků na jednotlivé páry pro JMODE = 0	51
Tabulka 9.1	Analýzovaný obsah ILAS	64

ÚVOD

Pro záznam a následnou analýzu rychlých dějů je třeba použít odpovídající rychlost vzorkování tohoto signálu. Při potřebě rychlého vzorkování, nad zhruba 10 – 100 MSa/s, narážíme na omezenou rychlost procesorů, které z principu činnosti nemají potřebnou rychlost pro obsluhu takto rychlých AD převodníků a následné ukládání dat.

V rámci této práce je představeno řešení spočívající v nasazení obvodu FPGA. Obvody FPGA jsou schopny z podstaty principu funkce poskytnout dostatečnou rychlost pro obsluhu rychlých AD převodníků a následného ukládání. Současně jsou schopny obvody FPGA nabídnout i obsluhu pomalejších částí systému. Pomalejší částí systému je v tomto případě míněno například odesílání naměřených dat do počítače s využitím rozhraní Ethernet, periferie typu SPI nebo uživatelská obsluha.

Pro účely této práce byly vybrány komerčně dostupné moduly – vývojová deska s A/D převodníkem firmy Texas Instruments a samostatná vývojová deska založená na obvodu FPGA rodiny Kintex 7 firmy Xilinx. Využití komerčně dostupných modulů dokáže dostatečně ohodnotit vlastnosti systému a v daném množství vychází při takto specializovaných obvodech jejich použití mnohem levněji než náklady spojené s vývojem a potřebnou výrobou prototypů. Jako komunikační rozhraní mezi FPGA a A/D převodníkem je použito JESD204B jako moderní standard vysokorychlostního přenosu s obvody ADC a DAC. Pro odesílání dat do počítače pak rozhraní Ethernet, který disponuje, oproti rozhraním typu RS232 či RS485 dostatečnou rychlostí. Práce shrnuje rovněž analýzu dnešního trhu, která ukázala poměrně omezené možnosti z pohledu dnešní součástkové základny (říjen 2017).

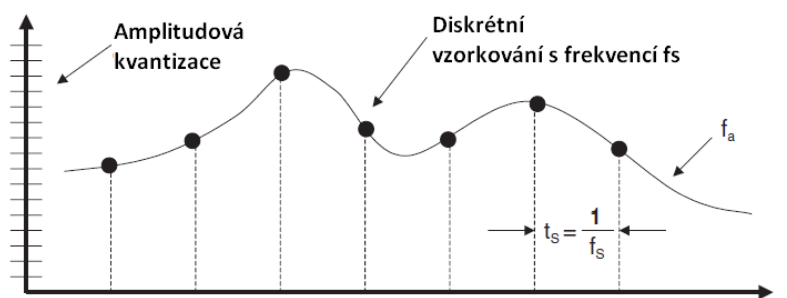
Práce je rozdělena do osmi kapitol. V prvních pěti kapitolách jsou shrnuty všechny důležité teoretické poznatky, ať se jedná o teorii A/D převodu, problematiku pamětí nebo použitých obvodů FPGA. Převážná část je pak věnována samotnému rozhraní JESD204B. Teoretická část se snaží směřovat na dnes používané trendy. Šestá kapitola této práce se už zabývá samotným návrhem hardwarové části práce – od přehledu trhu, až po technické řešení měřicího řetězce, včetně případných omezení nebo možných problémů. V kapitole sedmé je řešena implementace do obvodu FPGA, včetně implementovaného procesoru Microblaze, převážně se jedná o blok JESD204B, blok transportní vrstvy, blok ukládání a lwIP stack či SPI pro případ Microblaze. Informace o aplikaci pro příjem, zápis a vizualizaci dat jsou zmíněny v kapitole osmé. Testování, ať už při ožívování ADC nebo vyhodnocování změřených signálů je věnována kapitola osmá. Práce je ukončena závěrem, kde jsou shrnuty dosavadní výsledky, včetně možných vylepšení a další předpokládané práce, případně kritické zhodnocení navržených řešení.

1 A/D PŘEVOD

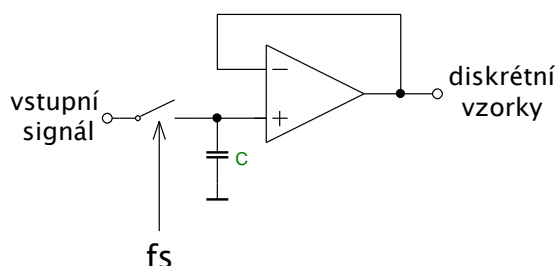
Veškeré fyzikální děje v přírodě jsou spojité, z pohledu elektrotechniky tedy analogové. Pro zpracování dat, v dnes běžných, digitálních systémech je nutné data digitalizovat, a to při minimálním zkreslení nebo jakékoliv degradaci aditivním šumem.

1.1 Průběh A/D převodu

Prvním procesem A/D převodu je vzorkování, při vzorkování nastává diskretizace signálu v časové oblasti – s rychlostí vzorkování jsou odebrány vzorky, které jsou dále zpracovávány (kvantovány) – viz obrázek 1.1. Vzorkování je zajišťováno obvodem Sample and Hold (SH), který zajistí odebrání vzorku a udržení jeho hodnoty po celou dobu převodu, ideově se jedná o zapojení kondenzátoru a následného bufferu s vysokou vstupní impedancí (napěťový sledovač), takže následný obvod nevybíjí kondenzátor. Ideové zapojení tohoto obvodu je na obrázku 1.2. Z hlediska vstupní kapacity je třeba reálně využít sofistikovanější zapojení [1]. Oproti vzorkování typu SH existuje ještě přístup typu TH (Track and Hold), kdy je spínač sepnut dříve, než je další vzorek požadován. V tomto případě pak je vstupní napětí určitou dobu sledováno.



Obrázek 1.1 Průběh vzorkování a kvantování při A/D převodu, převzato z [41]



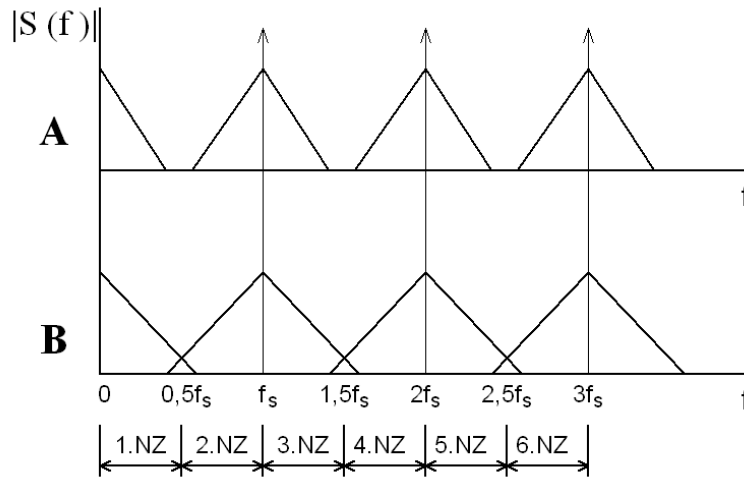
Obrázek 1.2 Jednoduchý obvod typu Sample and hold

Obvod zpracování musí zpracovat vzorek do času $1/f_s$, tedy do doby odebrání dalšího vzorku. Takto vzorkovaný signál může být teoreticky rekonstruován zpět na spojitý v jeho přesném obrazu, avšak pouze za dodržení Nyquistova teoremu – vzorkovací frekvence musí být minimálně dvojnásobkem nejvyšší frekvenční složky, která je v

signálu obsažena, tedy:

$$f_{VZ} \geq 2 \cdot f_{MAX}. \quad (1)$$

V případě, že tato podmínka splněna není, může dojít k překrytí jednotlivých spektrálních složek – některé frekvenční komponenty zasáhnou do dalších Nyquistových zón a přezrcadlí se zpět do první Nyquistovy zóny. Tím dojde ke zkreslení signálu. Situace je ukázána na obrázku 1.3. V prvním případě byla respektována Nyquistova podmínka, v první Nyquistově zóně je spektrum samotného signálu. Ve druhém případě signál nesplnil Nyquistovu podmínku – nachází se až ve druhé zóně a v první zóně je nežádoucí komponenta – došlo k překrytí spekter. U pásmových signálů k tomu dojít nemusí, čehož se využívá při podvzorkování [2]. K zamezení aliasingu může být před převodníkem zařazen antialiasing filtr, které frekvenční komponenty mimo pásmo potlačí. Antialiasingový filtr je filtr typu dolní propust s mezním kmitočtem poloviny vzorkovacího kmitočtu, případně menším.



Obrázek 1.3 Znárodnění Nyquistových zón pro dostatečnou vzorkovací frekvenci (A), pro malou vzorkovací frekvenci (B)

Každému takto odebranému vzorku je přiřazena konečná hodnota. Počet kvantizačních úrovní pro N-bitový převodník je 2^{N-1} . Úrovně jsou od sebe rovnoměrně vzdálené (vyjma případných nelineárních převodníků). Hladina pro rozhodování mezi sousedními hladinami je 50 %. Rozdíl mezi skutečnou úrovní vzorku a přiřazenou hladinou je kvantizační chyba, v případě digitálního vyjádření nabývá hodnoty maximálně $\pm 0,5$ LSB. Hodnotu 1 LSB lze vyjádřit jako [3]:

$$1 \text{ LSB} \geq \frac{FSR}{2^N}, \quad (2)$$

kde FSR je plný rozsah A/D převodníku ve V, což prakticky rovno referenčnímu napětí, případně U_{REF}/G , kde G je zisk vstupního zesilovače.

V důsledku kvantování vzniká kvantizační šum, což lze pokládat za superponovaný signál způsobený rozdílem mezi skutečnou hodnotou signálu a přiřazenou hodnotou signálu po kvantování.

1.2 Typy A/D převodníků

Flash A/D převodníky – v principu obsahuje 2^N rezistorů a 2^N-1 komparátorů. Referenční napětí je odstupňováno rezistory R. Převod probíhá v jednom taktu. Jsou rychlé, oproti tomu je třeba k realizaci velký počet komparátorů, další nevýhodou je poměrně vysoká vstupní kapacita a vysoká spotřeba. Jejich nabídka na trhu je dnes již silně omezená.

S řetězeným zpracováním – využívá metody zřetězení (pipelining), převod zde probíhá v několika oddělených částech. Hlavní myšlenka je v tom, že v okamžiku, kdy byla uvolněna první fáze, tak již může začít zpracování dalšího vzorku. Výhodou tohoto typu převodníku je poměrně velká rychlost. Nevýhodou je latence – první převod je uskutečněn za N fází, z principu činnosti dosahují vyšší chyby. Dnes nabízí vysoké rozlišení (až 16 bitů) s rychlostí nad 100 MSa/s. Tato architektura je pro vysokorychlostní převodníky využívána nejčastěji. Pro ještě vyšší rychlosti, v řádu GSa/s se běžně využívá několik jader na jednom čipu.

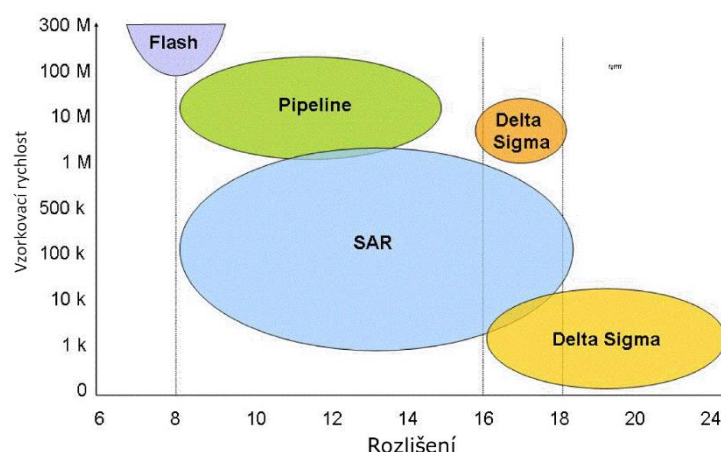
S interpolací – v případě převodníků typu Flash roste dramaticky potřebná plocha na čipu i spotřeba se zvyšujícím se rozlišením. Převodníky typu pipeline naopak potřebují několik fází pro převod. Převodníky s metodou interpolace využívají převodník typu flash s nižším rozlišením a následně aplikují speciální techniky interpolace [4].

S postupnou aproximací – výsledek převodu je převeden zpět na analogovou hodnotu D/A převodníkem, porovnán komparátorem a opět zpřesněn. Převod trvá několik cyklů. Jsou pomalé, oproti tomu mají malou spotřebu.

S dvojitou integrací – v první fázi (1. integrace) je na vstup přivedeno vstupní napětí, N-bitový čítač impulsy do přetečení. Po přetečení je na vstup přivedeno referenční napětí opačné polarity a je spuštěna druhá integrace. Napětí na výstupu integrátoru se zmenšuje k nule, v okamžiku, kdy dosáhne nuly je v čítači přímo digitální vyjádření vstupního napětí. Tyto převodníky jsou využity například v rámci multimetrů.

Sigma delta – založeny na rychlém vzorkování signálu, kde rychlost vzorkování je několikanásobně vyšší než požaduje Nyquistův teorém, na výstupu je pak digitální filtr, který průměruje výstupní signál a provádí decimaci. Výhodou je velmi vysoká linearita při velkém bitovém rozlišení (např. 24 bitů). Oproti tomu jsou vhodné spíše pro nižší frekvence (desítky kHz).

Na obrázku 1.4 je základní porovnání využití třech typů A/D převodníků pro rozlišení a rychlost vzorkování (rok 2007). Zatímco sigma-delta převodníky dosahují vysokého rozlišení, jejich rychlost je poměrně omezena. Zřetězenou architekturou (Pipeline) lze dosáhnout poměrně velké vzorkovací rychlosti, avšak nižšího rozlišení. Převodníky s postupnou aproximací pak poskytují kompromis mezi bitovým rozlišením a rychlostí vzorkování.



Obrázek 1.4 Využití A/D převodníků z hlediska rychlosti a rozlišení, převzato z [42]

1.3 Vybrané parametry A/D převodníků

A/D převodníky jsou kromě samotného rozlišení charakterizovány desítkami parametrů, ne vždy musí být u daného převodníku uvedeny všechny. S tím, jak je převodník více specifický roste i počet parametrů, jimiž je popsán. Obecně lze parametry rozdělit na statické (DC) a dynamické (AC). Zatímco statické parametry popisují převodník pro pomalé změny na vstupu, dynamické popisují parametry při rychlých změnách na vstupu.

Níže je zaměřena pozornost pouze na vybrané parametry, které mají bližší vztah k řešené problematice. Hlediska, podle kterých je A/D převodník vybírán je mj. ovlivněn aplikací, kde bude použit. Samotná otázka spotřeby není v řešené aplikaci důležitá, nicméně vysokorychlostní převodníky mohou dosahovat spotřeby jednotek wattů.

1.3.1 Efektivní počet bitů

Žádný dostupný A/D převodník není ideální. Jakýkoliv A/D převodník přidává k vstupnímu signálu šum, A/D převodník má dále vstupní nelinearitu, chybu v nastavení zisku nebo offsetu, což vede ke zkreslení signálu. Veličina Efektivní počet bitů, neboli ENOB (Effective number of bits) vyjadřuje počet bitů idealizovaného A/D převodníku, který by měl stejné rozlišení jako obvod zatížený šumem a zkreslením. Ideální tedy chceme, aby veličina ENOB korespondovala s udaným rozlišením převodníku. ENOB je definován [5] vztahem (3), resp. (4) a je definován přes veličinu SINAD (odstup signálu od šumu a zkreslení), která oproti veličině SNR (poměr signál/šum) uvažuje i zkreslení. Frekvence signálu má mj. vliv na vnitřní nelinearity ADC, a tak může být veličina ENOB frekvenčně závislá [6].

$$ENOB = \frac{SINAD_{meas} - 1,76 + 20 \cdot \log\left(\frac{U_{FS}}{U_{IN}}\right)}{6,02}, \quad (3)$$

kde U_{FS} je napětí plného rozsahu ve V a U_{IN} vstupní napětí ve V.

$$ENOB = \frac{SINAD_{\text{meas}} - 1,76}{6,02}, \quad (4)$$

1.3.2 Rychlost vzorkování a šířka pásma

Rychlost vzorkování je vyjádřením počtu vzorků za čas (mimo převodníky typu sigma-delta). Je udávána v jednotkách MSa/s či MSPS, apod. V rámci jednoho A/D převodníků může být nastavitelná. Je poměrně podstatné vědět, zda se jedná přímo o frekvenci, kterou je řízeno vzorkování nebo je této rychlosti dosaženo prokládáním (viz kapitola 1.4) z více jader, případně více diskretních A/D převodníků pokud je myšlen již určitý HW systém.

Vysokorychlostní A/D převodníky dosahují vzorkovací rychlosti až jednotek GSs/s při použití prokládání. Lze najít i informace o A/D převodníku s rychlostí 56 GSa/s [7] a více, zde je použito již pole stovky ADC pro získání výsledné rychlosti.

S rychlostí vzorkování částečně souvisí i parametr šířka pásma, ta je definovaná pro útlum signálu na vstupní analogové části o 3 dB, až do této maximální frekvence tedy bude útlum vždy nižší než 3 dB. Bývá měřen pro hodnotu, kdy výstupní rekonstruovaná hodnota klesne oproti vstupní (v plném rozsahu) právě o 3 dB. Označován často jako FPBW (Full Power BandWidth). Šířka pásma bývá obvykle vyšší než vzorkovací rychlost, nicméně situace je poměrně komplikovanější v případě, kdy je výsledné vzorkovací frekvence dosaženo již zmíněným prokládáním z více kanálů zde to platit nemusí – vstupní signál o určité frekvence je přiveden paralelně do více kanálů, které se na prokládání podílí, ale každý vstup vnitřního kanálu je omezen právě šířkou pásma. Rychlost vzorkování tedy sice roste, ale šířka pásma zůstává stejná, resp. se nezvyšuje.

1.3.3 Dynamický rozsah bez IM zkreslení

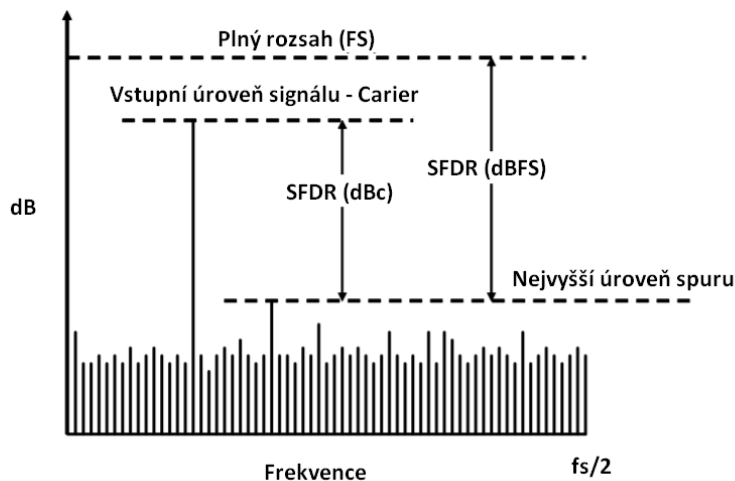
Dynamický rozsah bez intermodulačního zkreslení neboli SFDR (spurious free dynamic range) definuje dynamický rozsah mezi střední úrovní výkonu nosné a střední úrovní výkonu druhého nejsilnějšího (nežádoucího) signálu ve frekvenční doméně (např. FFT) tak, jak je zobrazeno na obrázku 1.5. V tomto rozsahu se tedy nesmí vyskytovat žádné jiné nežádoucí signály. Parametr je vyjádřen buď v jednotkách dBc, což vyjadřuje výkon vztažený k nosné nebo v jednotkách dBFS, což je hodnota signálu při plném rozsahu. Parametr SFDR může být definován následovně [8]:

$$SFDR = 20 \cdot \log \left(\frac{A_{\text{FUN,rms}}}{A_{\text{LSPUR,rms}}} \right), \quad (5)$$

kde $A_{\text{FUN,rms}}$ je výkon nosné a $A_{\text{LSPUR,rms}}$ výkon nejsilnějšího nežádoucího signálu.

U kvalitních převodníků je SFDR vyčleněn mezi první a druhou (případně třetí) harmonickou. V případě, že je AD převodníků určen pro úzké pásmo a druhá (třetí) harmonická je mimo toto pracovní pásmo, pak je SFDR definován v tomto úzkém pásmu. Ve druhém případě je definován širokopásmově. V případě, že se A/D převodníky skládají z více jader pro dosažení větší rychlosti vzorkování, se mohou ve spektru objevit rušivá rezidua vzniklá prokládáním, proto by mělo být u vícejádrových převodníků uvedeno, zda tyto komponenty byly uvažovány nebo ne [9].

Hodnota SFDR může být degradována jednak velkou úrovní šumu, kdy hodnota šumu bude v hodnotícím pásmu větší než samotná druhá či třetí harmonická, dále prokládáním, diferenciálním vstupem a následným přizpůsobením na single-ended vedení. Obvyklé hodnoty SFDR přesahují u kvalitních převodníků 50 dBFS.



Obrázek 1.5 Znáornění pojmu SFDR ve spektru, převzato z [5]

1.3.4 Další důležité parametry:

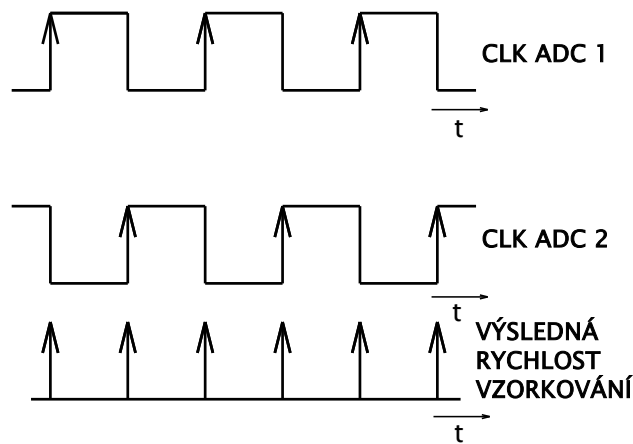
- INL [LSB] – integrální nelinearita – odchylka skutečné charakteristiky od ideální charakteristiky, typicky desetiny až jednotky LSB
- DNL[LSB] – diferenciální nelinearita – šířka kódů pro jednotlivé hodnoty vstupního napětí, typicky desetiny až jednotky LSB
- Offset [V] – vstupní offset napětí na vstupu A/D převodníky, většinou lze kalibrovat přímo v rámci ADC, v závislosti na převodníku se jedná o jednotky až stovky μV až mV.
- Offset drift [$\mu\text{V}/^\circ\text{C}$] – změna offsetu s teplotou
- Gain error [%FSV, LSB, V] – chyba nastavení zisku, odchylka od sklonu ideální přenosové funkce
- Gain drift [ppm/ $^\circ\text{C}$] – změna zisku, resp. útlumu s teplotou
- SNR, SINAD [dB] – odstup signálu od šumu bez uvažování zkreslení (SNR), s uvažováním zkreslení (SINAD)
- Odchylka od zpoždění mezi vzorky, spotřeba, způsob přepínání kanálů, potřebná doba k vnitřnímu nastavení, ...

1.4 Zvyšování vzorkovací frekvence

Možností, jak dosáhnout vyšší vzorkovací rychlosti systému je prokládání. Pro dosažení vzorkovací rychlosti m je použito několik A/D převodníků (pro diskrétní verzi většinou 2 nebo 4), kde každý vzorkuje rychlostí $1/m$. Taktovací signál pro jednotlivé převodníky je fázově posunutý. Pro případ 2 A/D převodníků je signál CLK posunutý o 180° (obrázek 1.6), pro případ 4 A/D převodníků o 90° , může se však jednat i například o prokládání za použití tří A/D převodníků. Například v [10] je struktura 10GSPS ADC, který pro dosažení výsledné vzorkovací frekvence využívá celkem 8 sub-A/D převodníků. Fáze ϕ_n konkrétního převodníku je dána [11]:

$$\phi_n = 2 \cdot \pi \cdot \left(\frac{n-1}{m} \right), \quad (6)$$

kde n je daný převodník a m je celkový počet převodníků, které se účastní prokládání.



Obrázek 1.6 Princip prokládání při dvou ADC

U všech použitých A/D převodníků musí být zajištěn stejný zisk, offset, zpoždění mezi náběžnou hranou taktovacího signálu a obvodu sample and hold a především – je třeba zajistit minimální (ideální nulový) posuv příchodu hodinového signálu (při uvažování konstantního fázového posunu) mezi jednotlivými převodníky – tzv. clock skew [11], [12]. Pro případ velmi rychlých převodníků se jedná o jednotky či desítky fs. V případě nedodržení těchto požadavků dochází k degradaci signálu při růstu hodnoty nežádoucích spurů ve spektru. Vzor n , sinusového signálu o frekvenci f_{in} vzorkovaný frekvencí f_s s posuvem r_m (v tomto případě normovaný s hodnotou 0 – 1 k frekvenci f_s), lze vyjádřit jako [13]:

$$s_m(n) = \cos \left(\frac{2 \cdot \pi \cdot f_{IN} \cdot n}{f_s} \right) + 2 \cdot \sin \left(\frac{2 \cdot \pi \cdot f_{IN} \cdot r_m}{f_s} \right) \cdot \sin \left(\frac{2 \cdot \pi \cdot f_{IN} \cdot \left(n + \frac{r_m}{2} \right)}{f_s} \right), \quad (7)$$

Z něhož lze vidět, že v případě nulového posuvu je druhý člen roven nule a ke zkreslení teoreticky nedochází. Minimálního posuvu lze dosáhnout v případě použití

přímo na čipu, horší je pak situace při využití prokládání za použití ADC jakožto diskrétních součástek, toto lze doporučit jen při nižších vzorkovacích rychlostech.

Fázového posuvu lze teoreticky dosáhnout například uzpůsobeným úsekem vedením, resp. ovlivněním jeho elektrické délky. Pro generování výsledných taktovacích signálů s nízkým jitterem lze využít například obvody typu LMK04828, resp. LMX2582. Tyto a podobné obvody syntetizují výsledný vzorkovací kmitočet na základě přesného a stabilního oscilátoru s frekvencí v řádu desítek MHz. Zároveň umožňují tyto obvody generovat několik různých taktovacích signálů signálu, které jsou vzájemně frekvenčně svázány. Typicky se používají děličky (pro celočíselné násobky) a násobičky. Násobička je často použita jen jedna, děliček pak několik pro každý výstup samostatně. Jediným problémem zůstává fázový posuv. Z hlediska transceiverů na FPGA však nehraje významnou roli.

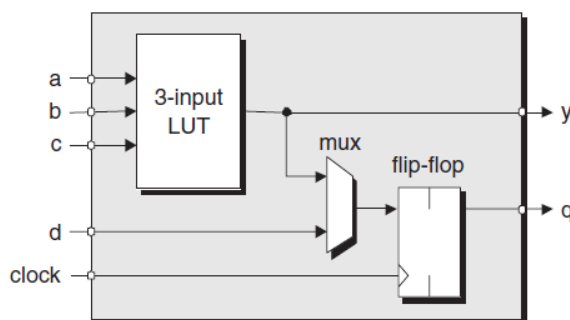
Především z hlediska implementace a zvyšování počtu jader, resp. kanálů na čipu může docházet ke zvyšování vstupní kapacity, a tím pádem ke snižování šířky pásma. Kapacita je mj. ovlivněna zvoleným typem obvodu SH, resp. TH [1].

2 OBVODY FPGA

2.1 Popis

Obvody FPGA (Field Programmable Gate Arrays) jsou digitální integrované obvody založené na matici programovatelných bloků které jsou vzájemně propojeny programovatelnou propojovací strukturou. Konfiguraci většiny dnes dominujících FPGA obvodů lze měnit, oproti tomu existují FPGA typu OTP, které lze nakonfigurovat pouze jednou. V souvislosti s obvody FPGA se lze setkat s obvody typu ASIC, obvody ASIC jsou tzv. zákaznické obvody, ty jsou vyráběny na zakázku pro specifickou aplikaci, obsahují přesně dané bloky potřebné pro dané využití.

V základu jsou obvody FPGA složeny z logických bloků (tisíce, statisíce). Například výrobci, rodinami nebo generacemi může být struktura rozdílná, v základě takový blok obsahuje několik LUT a flip-flop obvodů. Na obrázku 2.1 je základní struktura složená z jedné třívstupové LUT buňky, jednoho flip-flop obvodu a multiplexoru. V rámci výrobce Xilinx se využívá termínu slice, v dnešních FPGA obvodech je již struktura slice sofistikovanější.



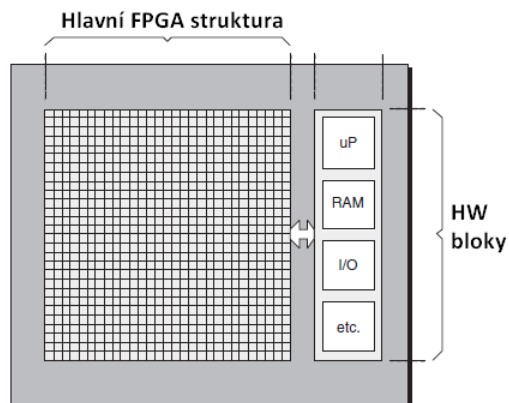
Obrázek 2.1 Stavba logické buňky obvodu FPGA, převzato z [44]

Dnešní FPGA bloky již obsahují i řadu specializovaných bloků. Pro použití těchto bloků pak není nutno používat základní strukturu FPGA (vyjma propojení). Jedná se například o blokovou RAM, bloky pro obsluhu vysokorychlostních vstupně-výstupních zařízení, bloky pro DSP operace, hard procesory, AD převodníky, apod. Situace je znázorněna na obrázku 2.2– samotná struktura FPGA vlevo, vpravo je pak skupina hardwarových bloků. Některé bloky lze vytvořit až „programově“, v tomto případě jsou však využity základní prostředky FPGA (soft procesory, distribuovaná RAM, apod.).

Pro případ obvodů firmy Xilinx se lze nejčastěji sekat se softprocesorem Microblaze, PicoBlaze nebo starším PowerPC. V případě hard procesorů se jedná o procesory ARM, těmi jsou vybaveny obvody rodiny Zynq. Ty se skládají mj. z části PL – programovatelné logiky a části PS (processing system), kde je implementován právě procesor ARM (Cortex-M).

Dalšími významnými a moderními HW bloky, které jsou dnes implementovány společně s FPGA jsou AD převodníky, lze se setkat s budoucími produkty, které budou vybaveny AD převodníky se vzorkovací rychlostí až řádu jednotek GSa/s, aktuální

rychlost implementovaných A/D převodníků se pohybuje v desítkách až stovkách MSa/s. Dále lze zmínit například bloky pamětí HBM či HBC, které již jsou implementovány v nejnovějších FPGA obvodech firmy Xilinx i Intel. Cena obvodů s pamětí HBM, resp. HMC je však velmi vysoká (stovky tisíc korun).



Obrázek 2.2 Struktura obvodu FPGA, převzato z [44]

U moderních a dnes vyvíjených obvodů typu FPGA se dnes používají výhradně pouzdra typu BGA, čímž lze vyřešit problém velkého počtu pinů při udržení malé plochy pouzdra. Počet pinů u nejnovějších obvodů (Virtex UltraScale+) přesahuje dva tisíce.

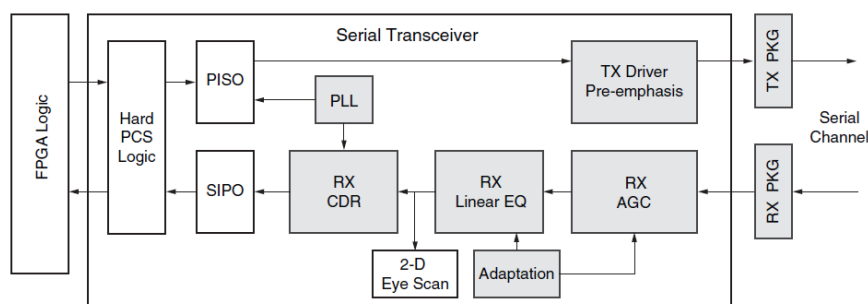
Nejznámější a největšími výrobci FPGA jsou Intel (dříve Altera) a Xilinx, dalšími výrobci jsou například Lattice Semiconductor nebo Microsemi. Cenové rozpětí obvodů FPGA je poměrně široké, v případě posledních generací se lze bavit o ceně v rozmezí řádu stokorun až po stovky tisíc korun v případě nejnovějších obvodů.

2.2 Vysokorychlostní transceivery

V případě potřeby rychlého přenosu dat (jednotky, desítky či stovky Gb/s) je třeba, v případě klasického paralelního přenosu dat běžným I/O rozhraním, velké množství datových vodičů. To jednak vyčerpává dostupné I/O piny FPGA a také komplikuje návrh (synchronizace napříč všemi vodiči, velké množství vodičů na desce s požadavkem stejné elektrické délky, apod.). K tomuto účelu slouží vysokorychlostní transceivery (GT). Příklad takového transceiveru je na obrázku 2.3. Jedná se o speciální vstupně-výstupní blok složený z bloků pro obnovu časování (CDR), blokem pro (de)serializaci, (de)kodérem 8/10, bufferem, bloky pro ekvalizaci (AQ) nebo AGC (automatic gain control). Výstupní budič je typu CML (current mode logic), kde lze i řídit vstupní, resp. výstupní impedanci pomocí terminačního rezistoru.

Ekvalizace je proces, který pomáhá vyrovnat vlastnosti přenosového kanálu a zvýšit tak schopnost detekce a obnovy signálu v přijímači. Obecně má přenosový kanál charakter dolní propusti. FPGA firmy Xilinx využívá ekvalizace DFE (Decision Feedback Equalization) a LPM (Linear Equalization). Typ ekvalizace je nutno volit také s ohledem na stav přenosového kanálu, rychlost přenosu a použitého rozhraní. Blokem, který souvisí s ekvalizací, a je ním také FPGA vybaveno, je blok pro snímání diagramu

oka, na základě tohoto diagramu kterého lze vyhodnocovat parametry kanálu a schopnost správné detekce a ladit tak nastavení vysílače, resp. přijímače. Vysílače používané ve spojení s vysokorychlostními transceivery jsou vybaveny také blokem pro přemfázi. Po začátku příjmu dat potřebují přijímač několik period k optimálnímu nastavení a správnému příjmu. V případě změny charakteru přenášeného signálu je pak někdy nutno nuceně vyvolat proces ekvalizace, v opačném případě nemusí dojít ke správné interpretaci dat.



Obrázek 2.3 Vnitřní struktura transceiveru, převzato z [45]

V tabulce 2.1 jsou příklady rychlostí a počtu dostupných transceiverů na obvodech FPGA firmy Xilinx. Jedná se o nejvyšší rychlost dostupnou v dané rodině. Konkrétní rychlost je značně závislá na konkrétním FPGA a jeho speed-gradu, tzn. že maximální rychlost se liší nejen v dané rodině, ale i na jednotlivých obvodech.

Tabulka 2.1 Možností transceiverů FPGA obvodů firmy Xilinx [14]

Rodina	Dostupné typy	Maximální rychlost [Gb/s]	Max. počet	Šířka pásma [Gb/s]
Virtex UltraScale+	GTY	32,75	128	8384
Kirtex UltraScale+	GTH/GTY	16,3/32,75	44/32	3268
Virtex UltraScale	GTH/GTY	16,3/30,5	60/60	5616
Kintex UltraScale	GTH	16,3	64	2086
Virtex 7	GTX/GTH/GTZ	12,5/13,1/28,05	56/96/16	2784
Kintex 7	GTX	12,5	32	800
Zynq UltraScale+	GTR/GTH/GTY	6/16,3/32,75	4/44/28	3268
Zynq 7000	GTX	12,5	16	400
Spartan 6	GTP	3,2	8	51

*Uvažována maximální dostupná rychlost napříč všem rodinám Xilinx

V případě rodiny Virtex UltraScale+ lze teoreticky dosáhnout maximální šířky pásma až 8384 Gb/s, oproti tomu například Spartan 6 nabízí maximální šířku pásma „jen“ 51 Gb/s.

2.3 Konfigurace obvodů FPGA

Po syntéze a implementaci designu je třeba patřičně obvod FPGA nakonfigurovat. Co se týká „programování“, resp. konfigurace obvodů FPGA neliší se prakticky od programování mikrokontrolérů. Na rozdíl od mikrokontrolérů však obvod FPGA nemá paměť pro program, tj. je po zapnutí vždy znovu konfigurován. Děje se tak přímým nahráním tzv. Bitstreamu nebo bývá konfigurace často uložena v paměťovém obvodu typu Flash nebo EEPROM, odkud je obvod FPGA po zapnutí nakonfigurován. Konfigurace může být i částečná, tzv. partial reconfiguration.

V případě, že je součástí FPGA designu také procesor, ať už hard či soft je nutno vytvořit tzv. bootloader. Pomocí bootloaderu je pak načten program procesoru.

Nejčastěji využívané rozhraní je JTAG. Pro připojení paměti Flash je využito rozhraní SPI. Další možností je konfigurace FPGA skrze Mastera, což může být například procesor, pomocí kterého je FPGA obvod nakonfigurován.

3 PAMĚTI

Pro rychlá měření, kde nelze data v reálném čase odesílat je nutné data ukládat a odeslání provést až po ukončení měření. Byť se v dnešní době již začínají na trhu objevovat FPGA obvody s pamětí typu HBM či HBC, tak stále je u většiny dnes dostupných FPGA obvodů paměť omezená – ať už se jedná o blokovou RAM či Ultra RAM. Většinou se jedná o jednotky až desítky Mbit v případě Blokové RAM (BRAM), případně desítek až stovek Mbit Ultra RAM. Z tohoto důvodu je nutno často použít externí paměť, a to jak o dostatečné velikosti, tak především dostatečné rychlosti.

3.1 Parametry paměti

3.1.1 Kapacita

Kapacita paměti vyjadřuje celkový počet paměťových buněk dostupných k uložení informace. Jedné buňce odpovídá jeden bit paměti. V případě paměti typu RAM se jedná o 2D strukturu, kde kapacita odpovídá součinu počtu řádků a počtu sloupců. Vyjadřována v násobcích bitů nebo bajtů.

3.1.2 Šířka sběrnice

Šířka sběrnice udává počet současně přenesených bitů. Při zachování stejného hodinového signálu lze tedy zvětšením šířky sběrnice výrazně dosáhnout větší šířky pásma, resp. datové propustnosti. Zvětšit šířku sběrnice je možné jak na úrovni diskretních paměťových obvodů, tak na úrovni hotových modulů (kde je výsledné šířky dosaženo právě spojením několika obvodů). V tomto případě však musí mít obvod (FPGA, mikrokontrolér, resp. procesor) dostatečný počet IO pinů.

3.1.3 Frekvence, rychlost

Frekvencí je řízen veškerý přenos dat z/do paměti. Frekvence hodinového signálu nemusí (ale může) korespondovat s rychlostí, resp. s počtem transferů z/do paměti. Příkladem jsou paměti typu DDRX, resp. QDR. Zde je rychlosti oproti hodinovému signálu dvojnásobná, resp. čtyřnásobná. Rychlost bývá často udáván přímo v b/s (a násobcích) nebo v jednotkách T/s, což vyjadřuje počet transferů za sekundu.

3.1.4 Šířka pásma

Šířkou pásma je míněná teoretická rychlost paměti bez jakékoliv režie. V podstatě ho lze vyjádřit jako součin počtu transferů a šířky sběrnice. V případě paměti typu DDR je tedy počet transferů dvojnásobek frekvence sběrnice. Šířky pásma prakticky nelze dosáhnout. Šířku pásma lze orientačně vyjádřit jako:

$$BM = w \cdot f \cdot p, \quad (8)$$

Kde BM je šířka pásma (Memory Bandwidth), w je šířka sběrnice vyjádřená jako počet bitů, f je frekvence taktovacího signálu a p vyjadřuje počet transferů během jedné periody, v případě DDR je tedy $p=2$, v případě QDR $p=4$.

3.1.5 Časové parametry

- CL: CAS Latency (Column Access Latency) [ns] – počet cyklů mezi adresací sloupce a odezvou. Prakticky doba, kdy je možné číst první bit paměti.
- TRCD: Row Address to Column Address Delay [ns] – počet hodinových cyklů mezi zpřístupněním řádků a přístupem do sloupce v tomto řádku.
- TRP: Row Precharge Time [ns] – počet hodinových cyklů mezi předběžnou adresací řádku a dobou, kdy je řádek připraven

3.1.6 Datová propustnost

Datová propustnost vyjadřuje skutečnou rychlost přenosu, ať už čtení či zápisu, které lze dosáhnout. Prakticky se jedná o šířku pásma sníženou o režii a jednotlivá zpoždění. Tato hodnota je silně závislá na způsobu práci s pamětí. Největší rychlosti přenosu se dosáhne při sekvenčním čtení, resp. zápisu, kdy jsou data zapisována postupně za sebe. V případě konkrétního paměťového obvodu se lze základně orientovat pomocí jeho speedgradu. V případě, kde je rychlost čtení nebo zápisu kritická je třeba se řídit benchmarky nebo tyto benchmarky provést a při provádění simulovat způsob práce s pamětí. Benchmark paměti prováděný na počítači nemusí vždy reflektovat limity paměti. Rychlost čtení, resp. zápisu může být větší.

3.2 Typy pamětí

3.2.1 DRAM vs SRAM

U pamětí typu DRAM (dynamic random access memory) je buňka tvořena jedním tranzistorem a jedním kondenzátorem. Kondenzátor se vybíjí a proto je nutno u těchto pamětí, pro zachování dat, provádět refresh. Vzhledem k tomuto faktu má tento typ paměti vyšší spotřebu a vysokou latenci, což způsobuje nízkou reálnou datovou propustnost. Cena těchto pamětí (vztahená na jeden bit) je nízká. Použití je především pro hlavní paměť, kde rychlost není kritická.

Oproti pamětem DRAM mají paměti typu SRAM (static random access memory) komplikovanější strukturu. Jako kondenzátor zde slouží parazitní kapacita použitých tranzistorů, tento kondenzátor se nevybíjí, z toho vyplývá, že data není třeba pravidelně obnovovat (refresh). Oproti DRAM je u těchto pamětí dosaženo velké šířky pásmy, nízké latence a tedy i vysoké reálné datové propustnosti. Cena těchto pamětí je oproti DRAM přirozeně vyšší, protože se používají jako rychlé paměti, kde není kritická velká kapacita.

3.2.2 Paměti typu DDRX SDRAM

Paměti typu DDR SDRAM (Double Data Rate Synchronous dynamic RAM, dále jen „DDR“) mají 2n prefetch architekturu uzpůsobenou k tomu, aby se vykonaly dva transfery během jednoho hodinového cyklu – na nástupnou a sestupnou hranu taktovacího signálu. Využívá se několik paralelních nediferenciálních vodičů (typicky 32, 64), které jsou společné pro zápis i čtení – jsou obousměrné.

U paměti typu DDR2 se opět vykonávají dva transfery během jednoho taktu. Využívají 4n prefetch architekturu. Oproti DDR není vyloučeno použití diferenciálních vodičů. Pro signálovou integritu je použita „one-die“ terminace, kde je přizpůsobení provedeno přímo na čipu, nikoliv na desce plošného spoje [15].

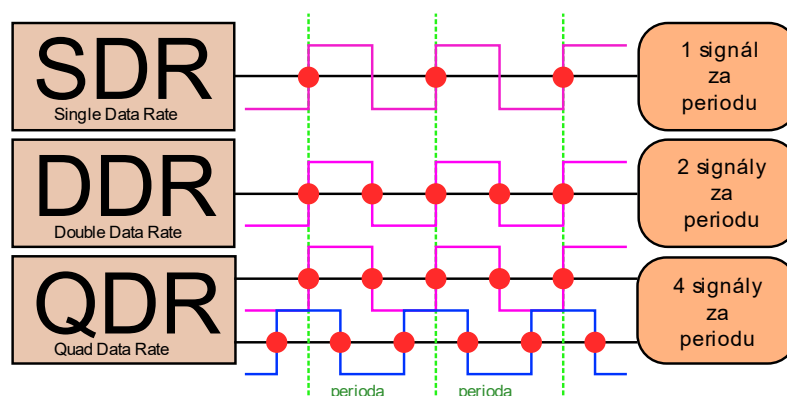
Třetí generace je označována DDR3, ty mají oproti pamětem DDR2 8n prefetch architekturu. Kromě „one-die“ terminaci je zde použita ještě „fly-by“ topologie [16], kde je routování provedeno sekvenčně – od jednoho DDR3 čipu k dalšímu, čímž se ještě více eliminují odrazy. Podobných technik je provedeno i u paměti DDR4, která má 16n prefetch architekturu. Paměti od třetí generace jsou způsobeny na práci s bursty.

3.2.3 Paměti typu QDR

QDR paměť (Quad Data Rate) má nezávislé porty pro čtení a zápis, u každého portu jsou navíc provedeny dva transfery během jednoho hodinového taktu. Tím je dosaženo toho, že paměť může běžet na čtyřnásobné rychlosti oproti taktovacímu signálu. Časový průběh časování se vzorkováním je na obrázku 3.1. Navíc u těchto pamětí nemůže dojít ke kolizi [16]. Operace s touto pamětí je orientovaná na režim práce s bursty (obdobně jako paměti typu DDR3/4). Šířka sběrnic se u těchto pamětí pohybuje v rozmezí 8 bitů až 36 bitů, včetně šířky 3 bitů. Poměr rychlostí zápis/čtení se u těchto pamětí blíží jedné, navíc mají paměti poměrně velkou šířku pásma. Prakticky jsou využity i na některých dostupných vysokorychlostních A/D kartách.

Pro případ typu QDR II je zápis příkazu pro zápis na náběžnou hranu a na sestupnou hranu pak ihned následuje adresa. Latence zápisu je tedy pro tento případ nulová.

Co se týče stavby paměti je uvnitř použit obvod DLL (Delay Locked Loop), ten lze volitelně vypnout, avšak výkon, paměti se v tomto případě sníží [17]. Terminační rezistor lze nastavit v rozmezí 175 a 350 Ω .



Obrázek 3.1 Průběh přenosu dat u paměti QDR, převzato z [43]

3.2.4 Paměti typu LPDDR

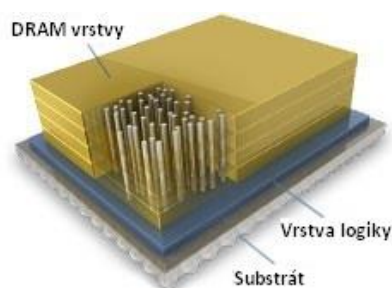
Oproti pamětem typu DDR mají paměti typu LPDDR nižší spotřebu. Toho je docíleno jednak použitím jiného napěťového standardu – 1,2 V (HSUL- 12) oproti standardům použitých u pamětí typu DDR (1,5/1,8, 2 V). Dalšího snížení spotřeby je dosaženo teplotní kompenzací, pomocí které nedochází k častému obnovování. Kromě toho, že se dnes paměti již tyto paměti využívají v mobilních zařízeních, mají podporu i pro použití s obvody FPGA.

3.2.5 Paměti typu RDRAM

RDRAM (Reduced Latency DRAM), jsou paměti DRAM, u kterých je dosaženo snížení latence a celkově zvětšení šířky pásma. Nízké latence a vysoké šířky pásma je dosaženo snížením doby náhodného přístupu (τ_{RC}), zjednodušením příkazů a využitím velkého počtu jednotlivých bank (například 16 pro případ RDRAM III). Rozdělení na jednotlivé banky snižuje parazitní kapacitu adresních a datových vodičů [16], což vede ke zrychlení přístupu a snížení případných kolizí. Schéma přenosu je stejné jako u pamětí typu DDR, tedy dva přenosy během jednoho taktu. Šířka sběrnice u obvodů paměti tohoto typu je 8 až 36 bitů.

3.2.6 Paměti HBM a HMC

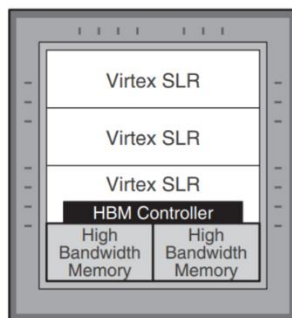
Jak paměti HBM tak paměti HMC tvoří, oproti pamětem typu RAM, 3D strukturu (obrázek 3.2), té je dosaženo pomocí vrstvení několika 2D struktur na sebe. V případě pamětí HBM je na sebe vrstveno několik (například 8) DRAM vrstev, které jsou vzájemně propojeny technologií TSV, logická vrstva je realizována jednou z vrstev nebo se jedná o samostatnou. V případě pamětí HMC je logická vrstva vždy jednou z vrstev. Oba dva typy pamětí již jsou pod standardem JEDEC. Paměti HBM a HMC nejsou vzájemně kompatibilní a zaměnitelné. V závislosti na generaci je dosahováno rychlosti desítek až stovek GB/s při udržení malé plochy čipu a malé spotřeby. Detailní informace těchto typech pamětech nejsou bohužel pro bližší popis dostupné.



Obrázek 3.2 Vnitřní struktura paměti typu HMC, převzato z [18]

Paměti HBM měly primární určení pro jednotky GPU a vývoj spadá pod výrobce Samsung, resp. AMD. Vývoj HMC v současnosti spadá pod výrobce Micron Technology. Z oblasti FPGA využívá výrobce Xilinx, stejně jako Intel paměti typu HBM. Tyto paměti jsou dostupné pouze v nejnovějších typech FPGA, z nichž mnohé nejsou stále dostupné

na trhu. Na obrázku 3.3 je ilustrována integrace pamětí typu HBM na obvodu XCVU37P Virtex UltraScale+. Paměť je napojena prostřednictvím HBM kontroleru na samotnou FPGA strukturu. Pro přístup ke kontroleru slouží AXI sběrnice, která po vylepšení umožňuje přenos až rychlostí 3,7 Tb/s [19] .



Obrázek 3.3 Integrovaná paměť HBM na FPGA XCVU37P firmy Xilinx, převzato z [19]

3.2.7 Porovnání

V tabulce 3.1 je porovnání pro vrcholné typy pamětí. Jak lze vidět, co se rychlosti týče, nemají paměti HBM a HMC konkurenci – konkrétně HBM 2 nabízí největší šířku pásma a malou energetickou zátěž.

Tabulka 3.1 Porovnání typů pamětí HBM, HMC, DDR4 a RLDRAM3 [19], [20]

Typ	HBM 2	HMC gen. 3	DDR4 - DIMM	RLDRAM3
Šířka pásma [GB/s]	460	160	21,3	12,8
Typ přístupu	DDR	Paketový	DDR	DDR
pJ/bit	7	30	27	40
Hloubka	8	4	16	5
Latence	Střední	Vysoká	Střední	Nízká

4 STANDARD JESD204

Standard JESD204B je standard vysokorychlostního rozhraní pro komunikaci mezi obvody FPGA a A/D, resp. D/A převodníky využívající vysokorychlostní transceivery. Lze se setkat i s DSP procesory (integrované s procesorem ARM), které toto rozhraní již nabízejí také. Komunikace mezi jinými zařízeními není teoreticky vyloučena. Jelikož je samotný standard [21] velmi obsáhlý, je popis rozhraní v této práci omezen pouze na základní fakta a informace nutné k pochopení problematiky a implementaci tohoto rozhraní.

4.1 Úvod

První standard JESD204 byl představen v roce 2006, jednalo se o přenos pomocí jednoho diferenciálního páru pracujícího v proudovém módu (CML). Rychlost přenosu byla v rozmezí 312,5 Mb/s až 3,125 Gb/s, volitelně. Časový signál byl společně rozveden pro převodník a přijímač, resp. vysílač. S rostoucí rychlostí a rozlišením A/D, resp. D/A převodníků roste potřeba přenosové rychlosti, a tedy i použití více párů. Tento problém částečně řeší první revize – JESD204A, kde je možno použít až 16 diferenciálních párů, celková přenosová rychlost systému je tedy až 16x větší.

Zatím poslední revize a dnes používaný standard je JESD204B (rok 2011). Tato revize jednak zvyšuje přenosovou rychlost na jednom páru (až na 12,5 Gb/s), ale především zavádí deterministickou latenci, tj. deterministické zpoždění mezi samotným odběrem vzorku až po zpracování v FPGA. Kromě deterministické latence také není u JESD204B použit jeden časový signál jako hlavní zdroj časování, ale každé zařízení může používat vlastní zdroj časování. Porovnání jednotlivých revizí je uvedeno v tabulce č. 4.1.

Pro určení latence může být využito několik mechanismů – pomocí signál SYSREF nebo SYNC, podle toho lze zařízení JESD204B rozdělit na tři podtřídy [11]. Zařízení třídy 0 (class 0) nemá mechanismus pro deterministickou latenci. Zařízení třídy 1 (class 1) využívá signál SYSREF a/nebo SYNC, třída 2 (class 2) pak signál SYNC. Třída 1 je primárně určena pro převodníky s rychlostí nad 500 MSa/s, třída 2 pak pro převodníky pod 500 MSa/s.

Tabulka 4.1 Porovnání revizí standardu JESD204X v kontextu s LVDS [11]

Typ	LVDS	JESD204	JESD204A	JESD204B
Rok	2001	2006	2008	2011
Maximální rychlost (1 linka)	1	3,125 Gb/s	3,125 Gb/s	12,5 Gb/s
Přenos po více linkách	NE	NE	ANO	ANO
Synchronizace napříč linkami	NE	NE	ANO	ANO
Synchronizace napříč zařízeními	NE	ANO	ANO	ANO
Deterministická latence	NE	NE	NE	ANO
Harmonické časování	NE	NE	NE	ANO

Počet využitých párů je libovolný (obvykle 1, 2, 4, 8 nebo 16), přenosovou rychlost lze rozložit za cenu použití více párů. Takovéto rozložení vede k ulehčení CDR, lepší synchronizaci, menší chybovosti a menšímu EMI. Aktuálně je ve vývoji JESD204C, který by měl používat kódování 64/66 s přenosovou rychlostí až 32 Gb/s. Níže jsou shrnuty výhody a nevýhody tohoto rozhraní [22]:

Výhody JESD204B:

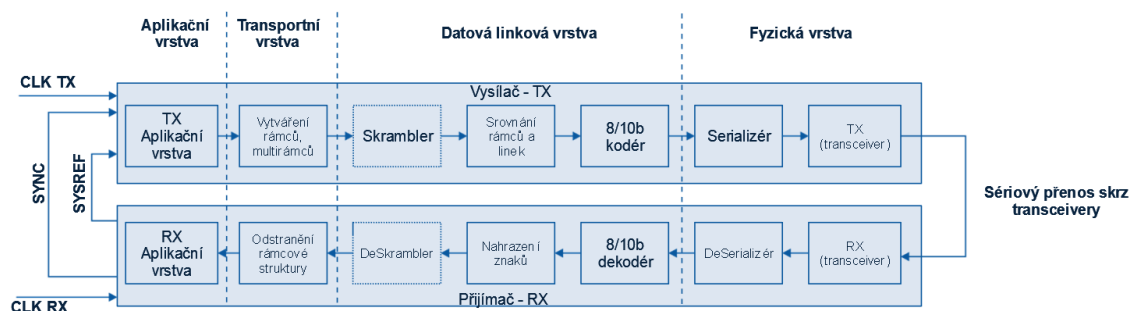
- Jednoduchý design PCB
- Menší plocha PCB
- Menší velikost pouzdra
- Škálovatelná rychlost (nelimitována setup/hold time)
- Zjednodušené časování a synchronizace

Nevýhody JESD204B:

- vyšší latence (vrstvení)
- Komplexnější firmware, složitější implementace, licencovaná IP jádra
- nutnost transceiverů

4.2 Vrstvy rozhraní

Na obrázku 4.1 je řetězec komunikačního rozhraní využívajícího rozhraní JESD204B. Rozdělení bloků do jednotlivých vrstev je poměrně neostré a v rámci různých popisů nebo produktu k tomuto rozhraní může být rozdělení poněkud odlišné. Pro popis se zaměříme na pohled přijímače, mj. právě přijímač bude realizován v rámci této práce.



Obrázek 4.1 Vrstvy a bloky rozhraní JESD204B

4.2.1 Fyzická vrstva

K přenosu dat se využívají speciální vysokorychlostní bloky – transceivery. Zařízení, vysílač – například A/D převodník je připojen k přijímači (například FPGA) pomocí několika diferenciálních párů, jejich počet je v terminologii JESD204 označen jako „L“ (lane). Při příjmu je třeba obnovit časování signálů, což zajišťuje blok CDR (Clock and

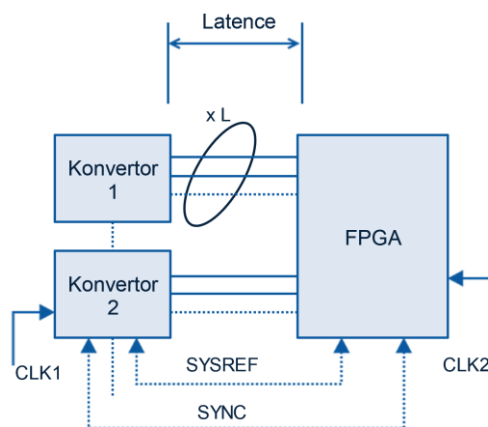
Data Recovery), provést deserializaci dat a dekodovat data (JESD204B využívá kódování 8/10). Všechny tyto bloky již jsou obsaženy ve struktuře transceiverů implementovaných na obvodech FPGA a není je tedy nutno zajišťovat. Podle využitých rychlostí (speed-gradu) jsou definovány parametry fyzické vrstvy – ty definují napěťové úrovně a časové parametry.

Tabulka 4.2 Možné speed-grady v rámci fyzické vrstvy JESD204B [23]

Speedgrade	1	2	3
Rychlost linky [Gb/s]	≤ 3.125	≤ 6.375	≤ 12.5
Výstupní diferenciální napětí [mV _{pp}]	500 (min) 1000 (max)	400 (min) 750 (max)	360 (min) 770 (max)
Doba náběžné/sestupné hrany [ps]	> 50	> 30	> 24
Výstupní jitter [ppUI]	0.35	0.30	0.30

Z tabulky 4.2 je patrné, že pro vyšší rychlosti se využívají nižší napěťové úrovně. Díky tomu je dosaženo větší rychlosti přeběhu, což vede ve výsledku k dostatečně otevřenému diagramu oka, a tedy snazšímu dekódování. Poměrně striktní časové parametry vedou k poměrně přísným pravidlům návrhu desky plošného spoje, především z hlediska udržení integrity signálu. Ve standardu JESD204 jsou mj. definovány právě parametry diagramu oka, které je třeba pro korektní funkci dodržet. Diagram oka je získáván měřením, například na vektorovém analyzátoru.

Celkově se rozhraní JESD204B skládá z celkem 4 (5) typů signálu. Fyzické provedení rozhraní je na obrázku 4.2. První signál jsou samotná přenášená data, jedná se až o až 16 diferenciálních párů (označovaných jako DA0+, DA0- až DB7+, DB7-), oproti samotných použitelných dat je tento datový tok zvětšen kvůli použití kódování 8/10. Dalším signálem je signál typu SYNC – skládá se ze signálu SYNCSE a diferenciálního páru TMSTP+ a TMSTP- (volitelně), tento signál slouží jako časová známka a slouží ke spuštění kódů pro zachycení synchronizace. Třetí signál CLK (diferenciální pár CLK+ a CLK-), což je samotný hodinový signál zařízení (např. A/D či D/A převodníku), slouží jak pro řízení vzorkování, tak pro řízení serializace. Poslední diferenciální signál je SYSREF (SYSREF+ a SYSREF-), což je systémová časová reference.



Obrázek 4.2 Fyzické provedení JESD204B

4.2.2 Datová vrstva

Datová vrstva přijímá od fyzické vrstvy paralelní data, provádí dekódování dat – není-li tato část řešena již fyzickou vrstvou v rámci transceiveru, nahrazuje řídicí znaky a volitelně provádí deskramblování. Pomocí řídicích znaků zajišťuje kontrolu možných chyb a vyvolávání případné resynchronizace. Práce datové vrstvy je poněkud odlišnější v případě skupinové synchronizace (CGS) a prvotní inicializace (ILAS), proto je inicializaci věnována zvláštní kapitola.

Pro účely zarovnání a detekci hranic rámců, resp. multirámců se využívají řídicí znaky, což jsou platné bitové sekvence. Pro účely rozhraní JESD204B se využívá 5 řídicích znaků: /K/, /F/, /A/, /R/ a /Q/, některé se využívají jen při samotné inicializaci. Bitové vyjádření jednotlivých symbolů je v tabulce 4.3, RD (Running Disparity) je dána poměrem počtu jedniček k počtu nul v předešlém slově. Použití jednotlivých znaků je následující:

- **/K/ = /K28.5/** – využívá se při CGS (skupinové) synchronizaci. Příjímač inicializuje žádost o synchronizaci signálem SYS do nízké úrovně. Po přijetí této žádosti začne vysílač vysílat znaky /K28.5/. V okamžiku, kdy přijímač přijme 4 po sobě jdoucí bezchybné znaky K28.5, nastaví signál SYS zpět do vysoké úrovně.
- **/A/ = /K28.3/** – tímto znakem je provedeno zarovnání multirámců. Je vkládán vysílačem na konec každého multirámce, v případě, že poslední oktet aktuálního rámce je roven poslednímu oktetu předešlého rámce – v tomto případě je poslední oktet aktuálního multirámce nahrazen tímto řídicím znakem.
- **/R/ = /K28.0/; /A/ = /K28.3/** – /R/ je použit pro označení začátku multirámce, /A/ označuje konec multirámce. V případě inicializační zarovnávací sekvence je znak /R/ první znak následující po znaku /K/ (viz výše). Použit je pouze při inicializační sekvenci.
- **/Q/ = /K28.4/** – používá se při první inicializační zarovnávací sekvenci, informuje přijímač o tom, že jsou zasílána konfigurační data JESD204B. Během samotného přenosu užitečných dat se nepoužívá.
- **/F/ = /K28.7/** – je vkládán vysílačem na konec rámce. V případě, že poslední oktet předchozího rámce je roven poslednímu oktetu aktuálního rámce, je tento oktet znakem /F/ nahrazen.

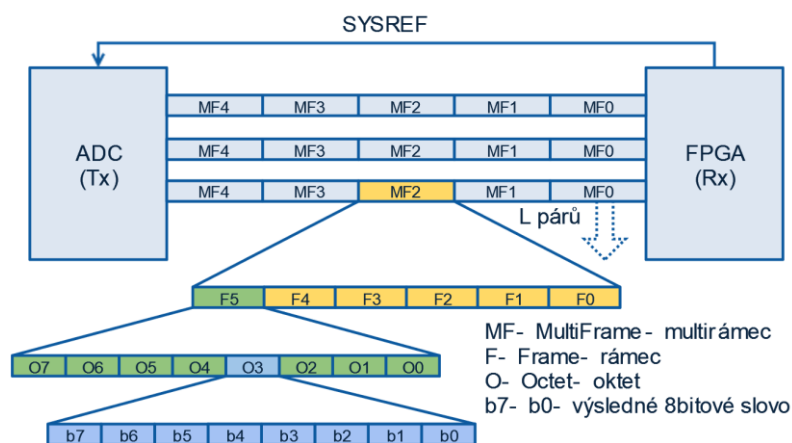
Tabulka 4.3 Řídicí znaky využívané rozhraním JESD204B [23]

Řídicí znak	Řídicí symbol	8bitová hodnota	10bitová hodnota pro RD = -1	10bitová hodnota pro RD = 1
/R/	K28.0	000 11100	00 1111 0100	11 0000 1011
/A/	K28.3	011 11100	00 1111 0011	11 0000 1100
/Q/	K28.4	100 11100	00 1111 0010	11 0000 1101
/K/	K28.5	101 11100	00 1111 1010	11 0000 0101
/F/	K28.7	111 11100	00 1111 1000	11 0000 0111

Scramblování zajišťuje nižší emise (EMI) do prostoru. Scramblování je volitelné a v žádném případě neprobíhá dříve než po ukončení fáze ILAS. Fáze CGS a ILAS jsou tedy vždy bez něj. Data jsou scamblována polynomm $1 + X^{14} + X^{15}$.

4.2.3 Transportní vrstva

Úkolem transportní vrstvy je vytvoření vhodné struktury data a předání těchto, již paralelních, dat datové vrstvě. Jelikož je náplní této práce obsluha A/D převodníku, zaměříme se na popis této vrstvy z pohledu přijímače. Operace ve vysílači jsou inverzní k operacím v přijímači. Transportní vrstva přejímá od vrstvy datové strukturu dat podobné na obrázku 4.3. Počet párů závisí na rychlostech použitých transceiverů, rychlosti a počtu bitů A/D převodníků a celkově na zvoleném módu přenosu. Každých 8 bitů tvoří oktet, oktety jsou vkládány do rámců a rámce do multirámců. Princip zapouzdření je zřejmý z obrázku 4.3.

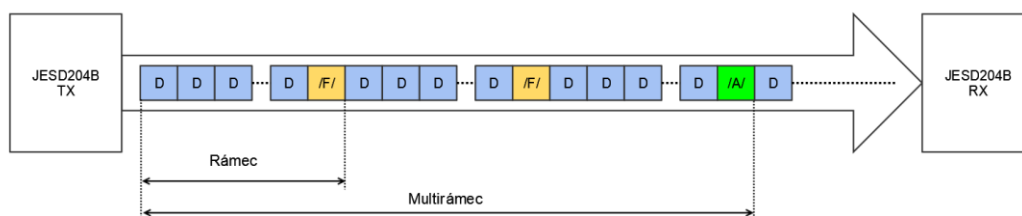


Obrázek 4.3 Základní struktury JESD204B

Počet bajtů F v rámci lze vypočítat jako [24]:

$$F = \frac{S \cdot N \cdot M}{8 \cdot L}, \quad (9)$$

kde S je počet vzorků během jedné rámcové periody, N je rozlišení převodníku, M počet převodníků v systému a L počet použitých párů.



Obrázek 4.4 Hranice rámce a multirámce v datovém toku

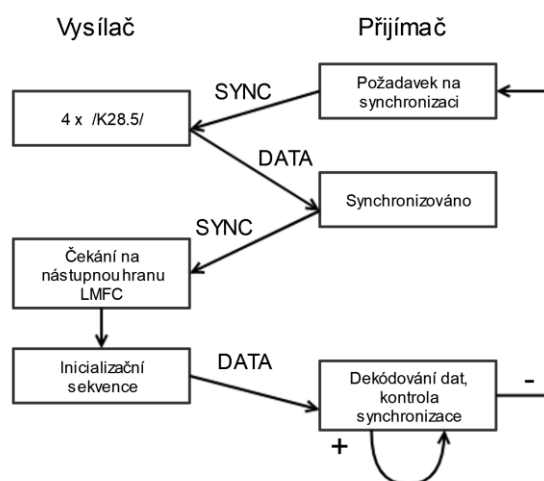
Počet oktetů na rámec, resp. počet rámců na multirámec není pevně daný standardem, ale je určen danou (uživatelskou) konfigurací použitého A/D převodníku. Nastavení parametrů je provedeno jak na straně vysílače (ADC), tak na straně přijímače (FPGA), během inicializace (viz předchozí kapitoly) dochází k ověření těchto parametrů. Přijímač může zasílané nastavení ignorovat nebo ověřit shodu se svým nastavením.

4.2.4 Aplikační vrstva

Kromě samotného poskytování surových dat (například vzorky získané A/D převodníkem) k následnému mapování nižší vrstvou zahrnuje aplikační vrstva i možnosti zvláštní konfigurace rozhraní, což může být jiné, než typické mapování dat. Nutné však je, aby tyto změny reflektoval i přijímač. Tuto možnost může A/D převodník využít ke zvýšení efektivity přenosu, případně ke snazšímu zpracování v aplikační vrstvě přijímače.

4.3 Skupinová synchronizace a inicializace

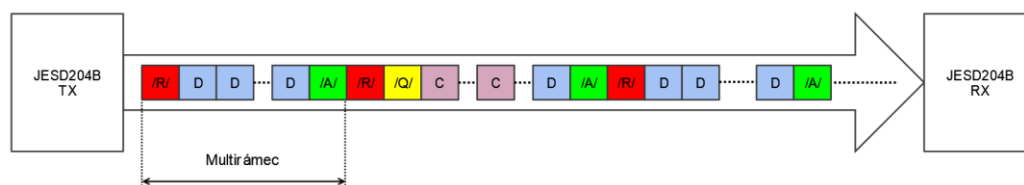
Prvním krokem, který předchází samotnému odesílání dat je fáze zvaná CGS (Code Group Synchronizace – kódová skupinová synchronizace). CGS je inicializována požadavkem přijímače na synchronizaci pomocí signálu SYSREF. Po tomto požadavku začne vysílač vysílat symboly K28.5 (viz výše). V okamžiku, kdy přijímač na všech párech přijme čtyři po sobě jdoucí znaky K28.5 deaktivuje signálem SYNC požadavek na synchronizaci. V případně úspěšně provedené fázi CGS začne vysílač s inicializační sekvencí ILAS. Postup je zjednodušeně (včetně ILAS) znázorněn na obrázku 4.5. LMFC je lokální multirámecový CLK signál (Local Multiframe Clock), touto hranou jsou vymezeny hranice multirámců.



Obrázek 4.5 Postup při CGS a ILAS

Během ILAS jsou detektovány hranice rámců, resp. multirámců a dojde k zarovnání dat na jednotlivých párech. ILAS se skládá minimálně ze 4 multirámců – struktura ILAS je na obrázku 4.6. Většina řídicích symbolů využitých během ILAS již při běžném zasílání dat není využita. První multirámec ILAS začíná znakem /R/ a končí znakem /A/, druhý rámec začíná opět znakem /R/, který je následován znakem /Q/ (znak /Q/ je využit pouze na tomto místě), po tomto znaku následují konfigurační data rozhraní. Třetí a čtvrtý

multirámec je totožný. Data obsažená mezi řídicími symboly mohou být přijímačem využita, případně jsou ignorována. V případě ADC mohou být v 1., 3. a 4. multirámcí zasílána data rampové funkce [23]. Po přijetí posledního znaku /A/ ILAS následuje již zasílání vlastních dat. V případě chyby může být vyvolána resynchronizace – ta je spouštěna datovou vrstvou, která mj. obstarává právě práci s řídicími symboly.



Obrázek 4.6 Inicializační sekvence v rozhraní JESD204B

4.4 Porovnání s LVDS

Následující tabulka 4.4 poskytuje porovnání rozhraní LVDS a JESD204 z pohledu fyzické vrstvy. I u velmi rychlých převodníků se lze setkat s rozhraním LVDS. Pokud nebereme v úvahu nutnost implementace samotného JESD204 rozhraní a zaměříme pozornost pouze na fyzickou vrstvu je rozhraní JESD204B mnohem výhodnější než LVDS. Efektivitu kódování nelze brát jako nevýhodu. Kromě tohoto JESD204B disponuje, jak již bylo zmíněno, deterministickou latencí.

Tabulka 4.4 Porovnání fyzické vrstvy rozhraní LVDS a JESD204B [25]

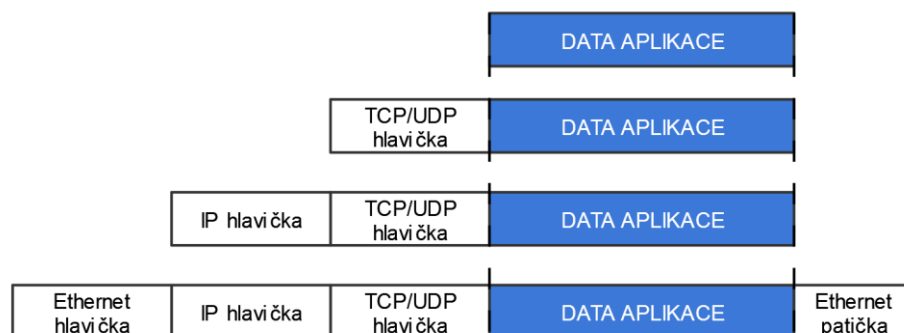
Typ	LVDS	JES204B
Počet spojů pro clock	2 (1 pár)	přímo v systému
Počet spojů pro data (příklad pro 16bit. 250 MSa/s ADC)	32 (16 párů)	1/2/4 páry – dle konfigurace
Požadované přizpůsobení	ano	ne
Shoda fáze při více ADC	komplikované	ano
Efektivita kódování	100 % (nepoužívá)	84 % (8/10B)

5 APLIKACE ETHERNETU

Standard Ethernet, definovaný normou IEEE 802.3 je jedna z realizací linkové a fyzické vrstvy referenčního modelu ISO/OSI, resp. vrstvy síťového rozhraní v případě modelu TCP/IP. Pro samotný přenos aplikačních dat je třeba však implementovat i zbývající nižší vrstvy, proto jsou v následující kapitole rozebrány.

5.1 Zapouzdření dat v síti TCP/IP

Na obrázku 5.1 je znázorněno postupné zapouzdřování dat v síti TCP/IP. Vyšší vrstva předává data vrstvě nižší, nižší vrstva doplňuje data o hlavičku, případně patičku, a poté výsledná data opět postupuje následující nižší vrstvě. Původní data aplikace zůstávají nezměněna, výsledný datový tok je však navýšen o režii jednotlivých vrstev. Nejvyšší vrstvou (L4) je vrstva aplikační (v rámci ISO/OSI modelu se jedná o vrstvy aplikační, prezentační a relační). Tato vrstva předává data vrstvě transportní, ta na základě zvoleného protokolu doplní patřičnou hlavičku a předává data vrstvě síťové, kde je hlavním parametrem IP adresa (IPv4, IPv6). Mezi vrstvou síťovou a samotným fyzickým médiem leží vrstva síťového rozhraní, zde je realizováno například rozhraní Ethernet. Adresování v této vrstvě probíhá pomocí MAC adresy. Na straně příjemce je pak postup získání aplikačních dat opačný. MAC adresa je dána fyzickým rozhraním a měla by být jedinečná. IP adresa je buď daná pevně nebo je získána skrze DHCP server.



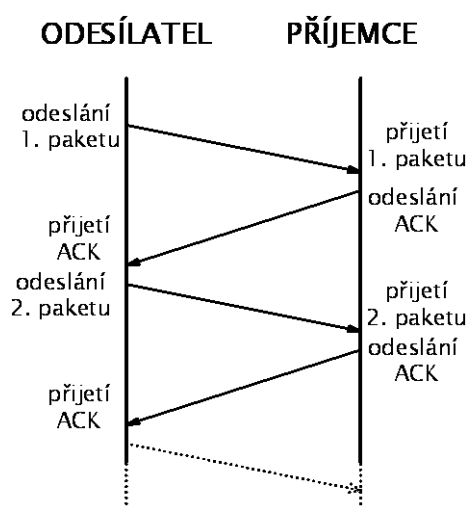
Obrázek 5.1 Zapouzdření dat v síti TCP/IP

5.2 Protokoly v síti TCP/IP

Přenos dat mezi dvěma „účastníky“ realizuje třetí vrstva modelu TCP/IP. Nejčastěji používanými protokoly jsou TCP (Transmission Control Protocol) a UDP (User Datagram Protocol). Vedle nich se vyskytují ještě určité méně známé protokoly, nejčastěji odvozené z předešlých dvou, například Multipath TCP, DCCP nebo UDP-Lite). V další části budou popsány pouze protokoly TCP a UDP.

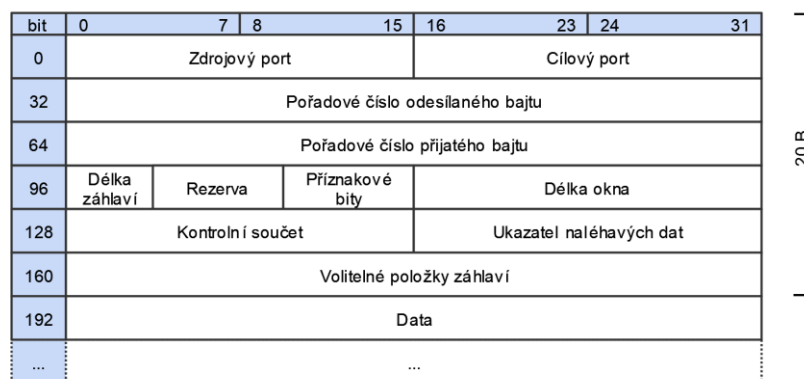
5.2.1 Protokol TCP

Protokol TCP realizuje spolehlivou spojovanou službu, to mj. znamená, že příjemce potvrzuje přijetí každé zprávy. Spolehlivosti je dosaženo potvrzováním přijatých dat (paketů), k čemuž slouží ACK. Na obrázku 5.2 je zobrazen průběh odesílání paketů od odesílatele k příjemci, v tomto případě je potvrzován každý přijatý paket zvlášť (metoda Stop and Wait). V případě, že není potvrzeno přijetí paketu (pomocí ACK) je paket po uplynutí Timeoutu odvyšlán znovu. Timeout musí být zvolen rozumně, v případě krátkého nastaveného timeoutu může být paket doručen dvakrát, jelikož doba přenosu ACK byla vyšší než nastavený timeout. Současně může dojít k doručení paketu dvakrát v případě, že sice první paket byl doručen, ale vlivem chyby již nebylo doručeno jeho potvrzení. Pakety nemusejí být odesílány postupně po každém přijetí ACK, pro zvýšení rychlosti lze pakety odesílat a potvrzovat po skupinách [26].



Obrázek 5.2 Odesílání paketů prostřednictvím protokolu TCP, metoda Stop and Wait

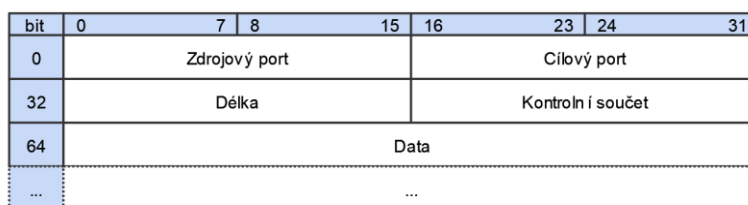
Bytestrem je pro účely protokolu TCP rozděluje na segmenty, společně s principem činnosti je záhlaví TCP protokolu oproti UDP poměrně složitější. Záhlavní je na obrázku 5.3. První 4 byty jsou opět vyhrazeny pro zdrojový port a cílový port. Následuje pořadové číslo odesílaného bajtu (4 B) a následujícího přijímaného bajtu (4 B), tato pořadová čísla jsou důležitá z hlediska principu TCP protokolu. Délka záhlaví vyjadřuje velikost TCP segmentu v násobcích 32 bitů. Příznakových bitů je 6 a mají různý význam (např. odmítnutí spojení). Délka okna vyjadřuje množství dat, které je příjemně schopen ještě přijmout. Klíčovou položkou je pak kontrolní součet k zabezpečení integrity přenášených dat.



Obrázek 5.3 TCP segment

5.2.2 Protokol UDP

Oproti protokolu TCP realizuje protokol UDP nespolehlivou nespojovanou službu, tzn. že příjemce nepotvrzuje přijetí zprávy, odesílatel se o přijetí zprávy nezajímá. Případné zajištění spolehlivosti může být dosaženo pouze koncovou aplikací (je-li to vyžadováno). Struktura UDP datagramu je zřejmá z obrázku č. 5.4. Záhlaví datagramu má velikosti 8 B, skládá se z 16 bitového čísla portu příjemce, 16bitové čísla portu odesílatele. Druhá polovina záhlaví je určena pro délku UDP datagramu (16 bitů) a kontrolní CRC součet (taktéž 16 bitů), ten je v případě UDP nepovinný. Je doporučeno používat datagramy o délce maximálně 512 B, čímž se zabrání případné segmentaci. Protokol UDP má malou přenosovou režii a je jednoduchý na implementaci – v podstatě se jedná o zabalení dat a jejich odeslání.



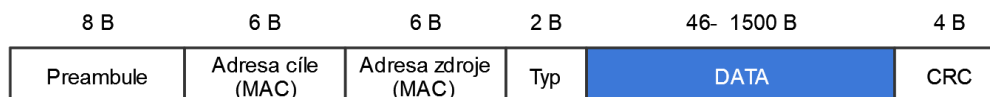
Obrázek 5.4 UDP datagram

5.3 Ethernet

Jednou z realizací vrstvy síťového rozhraní je právě Ethernet, přesněji standard IEEE 802.3. Z hlediska modelu ISO/OSI se jedná o vrstvu linkovou a fyzickou. Fyzickou vrstvou je samotné přenosové médium, např. metalický kabel. Vrstva linková je rozdělena na vrstvu MAC a vrstvu PHY. Vrstva MAC (Medium Access Cotnroler) řídí přístup ke sdílenému médiu. Vrstva PHY zajišťuje kódování, obnovu taktu, kompenzaci přeslechu, napěťové úrovně, apod. Pro komunikaci mezi vrstvou PHY se dnes nejčastěji používá rozhraní MII či RMII (redukované MII). Zatímco MAC vrstva je nejčastěji řešená přímo

implementací v rámci FPGA [27] či MCU, vrstva PHY bývá nejčastěji řešena externím obvodem. Lze však nalézt i mikrokontroléry, které v sobě mají vrstvu PHY integrovanou coby HW blok. Mimo jiné existují i specializované integrované obvody, které realizují celou vrstvu Ethernetu. Možnosti závisí i na potřebné dosažitelné rychlosti.

Na obrázku 5.5 je rámec Ethernetu. Blok DATA zde již představují data poskytnutá vyšší vrstvou (nikoliv tedy data aplikační vrstvy). Počátek rámce tvoří 8B synchronizační preamble. Poté následuje 6B MAC adresa cíle a 6B MAC adresa zdroje. Fyzická MAC adresa je (teoreticky) celosvětově unikátní. Skládá z identifikace výrobce (každý výrobce má přidělen rozsah) a z konkrétní identifikace v rámci daného výrobce. 2B pole typ je údaj o vyšším protokolu, případně je toto pole nahrazeno informací o délce či přesněji informací o odesílateli a příjemci. Datové pole má velikost 46 B – 1500 B, kdy 46 B je minimum, v případě menší velikosti je pole doplněno. Poslední 4 B tvoří hodnotu kontrolní součtu CRC, pomocí kterého lze detekovat, byl-li rámec při přenosu jakkoliv poškozen.



Obrázek 5.5 Rámec Ethernetu

V rámci verzí Ethernetu se lze setkat nejčastěji s rychlostí 10Mbit, 100Mbit a 1Gbit. Poslední standardizované rozhraní je prozatím 10Gbit Ethernet. Zatímco první tři verze používají metodu CSMA/CD k řešení kolizí na sdíleném médiu, verze 10Gbit již pracuje plně duplexně.

6 ŘEŠENÍ HW ČÁSTI

6.1 Vstupní požadavky

Od navrhovaného systému se očekává vzorkování rychlostí minimálně 4 GSa/s s hloubkou minimálně 8 bitů. Délka záznamu je zadavatelem práce stanovena na hodnotu 100 μ s. Požadován je rovněž přenos dat do počítače pomocí rozhraní Ethernet a následné ukládání do souboru typu CSV.

Z hlediska takto rychlého vzorkování není možno uvažovat o přenosu dat do počítače v reálném čase. Získané vzorky bude třeba průběžně ukládat do paměti a po skončení měření je do počítače odeslat. Doba záznamu je krátká a je tedy zbytečné začátek odesílání po spuštění měření. Uvažujeme-li minimální požadavky, vznikne požadavek ukládat data s rychlostí cca 10 GB/s, resp. 76,8 Gb/s. Pro stanovenou dobu záznamu 100 μ s je tedy požadovaná minimální velikost paměti 8 Mbit.

Obvody FPGA musí mít dostatek prostředků pro použití všech potřebných bloků (obsluha A/D převodníků, ukládání dat, následné čtení dat a obsluha rozhraní Ethernet), pro komunikaci s A/D převodníky po rozhraní JESD204B jsou třeba vysokorychlostní transceivery. Všechny tyto požadavky musí použitý obvod FPGA splňovat.

V neposlední řadě, bude-li použito komerčních modulů, je třeba zajistit, aby jednotlivé moduly šly snadno propojit bez potřeby redukcí nebo jiných „prostředníků“ ve formě zvláštní desky plošného spoje.

6.2 Analýza trhu a výběr komponent

Při výběru komponentů a analýze trhu byla pozornost zaměřena jak na dostupné hotové moduly, tak na diskrétní součástky. Zmapována byla také nabídka komplexních řešení. V následující kapitole jsou u jednotlivých komponent uvedeny přehledové tabulky, kde jsou pro přehlednost vybrány pouze určité parametry. Detailní přehled lze nalézt v rámci elektronické přílohy.

6.2.1 A/D převodníky

Požadavkům na rychlost vzorkování vyhověli, a to pouze částečně, jen výrobci Texas Instruments, Analog Devices a E2V. Ani v jednom případě nemá samostatný A/D převodník rychlost vyšší než 6,4 GSa/s. Za zmínku stojí také A/D převodník firmy Fujitsu, který má vzorkovací rychlost 56 GSs/s, nepodařilo se však nalézt detailnější parametry ani cenu. Jedná se však o matici A/D převodníků vytvořenou přímo na čipu. Architektury analyzovaných převodníků jsou typu pipelined v případě Analog Devices nebo s metodou interpolace za účasti komparačního ADC (Folding Interpolating) v případě Texas Instruments. Hodnota SFDR je závislá na frekvenci signálu a v rámci různých obvodů bohužel nelze uvést hodnotu pro stejné podmínky.

Tabulka 6.1 Přehled vyhovujících A/D převodníků pro požadované parametry

Označení	Výrobce	Rozlišení [bit]	Rychlost vzorkování [Gsa/s]	Šířka pásma [GHz]	SFDR [dBFS]	Rozhraní	Cena CZK
ADC12DJ3200	Texas Instruments	12	3,2/6,4	8	58 (2,5 GHz)	JESD204B	48400
LM97600	Texas Instruments	8	1,25/2,5/5	1,2	50 (1 GHz)	LVDS	5478
ADC12J4000	Texas Instruments	12	4	3,2	75 (0,7 GHz)	JESD204B	42900
AD9625	Analog Devices	12	2/2,5/2,6	3,2	57 (0,7 GHz)	JESD204B	40634
EV10AQ190A	E2V	10	1,25/2,5/5	3,2	56 dBc (1,6 GHz)	LVDS	17600
Fujitsu 65LL	Fujitsu	8	56	-	-	-	-

Pro případ, kdy nechceme využít prokládání z diskretních ADC vyhovují požadavkům pouze převodníky ADC12DJ3200, EV10AQ190A a LM97600. První zmíněný převodník dosahuje nejvyššího bitového rozlišení, nejvyšší hodnoty SFDR a používá rozhraní JESD204B. Těmto parametrům pak odpovídá i cena, která je ze všech analyzovaných převodníků nejvyšší. Zbylé dva převodníky dosahují nižšího rozlišení, převodník LM97600 pak také velmi malé šířky pásma. LVDS rozhraní by pak mohlo být také komplikované, zvláště z důvodu použití případných modulů, kde je poměrně velká vzdálenost mezi ADC a FPGA. Zvolen byl tedy ADC12DJ3200.

6.2.2 A/D moduly

Pro korektní funkci A/D převodníků a bezchybný přenos dat do FPGA je poměrně stěžejní částí návrhu celého systému samotný návrh desky, jedná se například o vhodné rozvržení vrstev, dodržení elektrických délek. Návrh takovéto desky vyžaduje značné zkušenosti a z hlediska počtu vrstev (např. 16) i dostatečně profesionální nástroje, výsledná výroba (několik obvodů v BGA pouzdrů) pak strojové osazování. Nelze ani vyloučit potřebné prototypy v případě odhalení chyb. Z časového hlediska je pak tedy hotový modul vhodnou volbou.

Tabulka 6.2 Přehled dostupných A/D modulů pro požadované parametry - výběr

Označení	Výrobce	Typ ADC	Výstupní konektor	Cena CZK
ADC12DJ3200EVM	Texas Instruments	ADC12DJ3200	HPC FMC+	55000
FMC126	Abaco Systems	EV10AQ190	HPC FMC 400 pin	220330
FMC125	Abaco Systems	EV8AQ160	HPC FMC 400 pin	neuvedeno
FSF-AD15000A	Fidus Systems	AD9625	2x HPC FMC 400 pin	57750
HTG-FMC-12ADC-16DAC	HitechGlobal	neuvedeno	HPC FMC 400 pin	neuvedeno
FMC161	Abaco Systems	ADC12D1800	HPC FMC 400 pin	neuvedeno

Označení	Výrobce	Typ ADC	Výstupní konektor	Cena CZK
EV10AQ190-EB	E2V	EV10AQ190A	neuvedeno	38390
FMC170	Abaco Systems	EV10AQ190	HPC FMC 400 pin	neuvedeno

Vývojové desky a A/D převodníky jsou vesměs osazeny obvody, které byly analyzovány v předchozí kapitole. V úvahu tedy připadá deska ADC12DJ3200EVM s A/D převodníkem ADC12DJ3200. Deska využívá standardizovaného konektoru FMC+ s možností redukce na FMC. Oproti samotnému obvodu je pak cenový rozdíl vůči kompletní desce zanedbatelný.

6.2.3 Obvody FPGA

V případě obvodů FPGA byla pozornost zaměřena na výrobce Xilinx, především pak na rodiny sedmé generace a vyšší. Každá rodina obsahuje několik různých obvodů, které se liší počtem pinů, transceiverů, typem použitých transceiverů, LUT buněk, bloků pro DSP zpracování, clock managerů a velikostí paměti. Vybrané obvody rodiny Virtex/Kintex UltraScale+ již obsahují paměť typu HBM. Možných konkrétních FPGA obvodů jsou stovky, do přehledu proto byly zahrnuty pouze některé.

Tabulka 6.3 Přehled vyhovujících FPGA obvodů - výběr

Označení	Rodina	Paměť			Diff IO	Single IO	GT	Cena CZK
		BRAM [Mb]	Ultra [Mb]	HBM [GB]				
VU37P	Virtex US+	70,9	270	8	-	624	96	-
XCVU13P	Virtex US+	94,5	360	0	414	624	128	932404
XCKU3P	Kintex US+	12,7	13,5	0	288	304	16	25014
XCKU15P	Kintex US+	34,6	36	0	624	668	76	82522
XCKU025	Kintex US	12,7	0	0	144	312	12	21868
XCKU115	Kintex US	75,9	0	0	384	832	64	119438
XCVU065	Virtex US	44,3	0	0	240	520	40	119724
XCVU440	Virtex US	88,6	0	0	1456	672	48	792176
XC7V585T	Virtex-7	28,6	0	0	408	850	36	76538
XC7V485T	Virtex-7	37	0	0	336	700	56	81180
XC7VX690T	Virtex-7	52,9	0	0	480	1000	80	122804
XC7K70T	Kintex-7	4,8	0	0	144	300	8	2926
XC7K325T	Kintex-7	16	0	0	240	500	16	20548
XC7K480T	Kintex-7	34,3	0	0	192	400	32	75548
ZU25DR	Zynq US+ RF SoC	27,8	13,5	0	-	152	12	-
ZU9CG	Zynq US+	32,1	0	0	-	328	24	55022

V případě, že bude pro uložení vzorků využito paměti FPGA, je třeba volit takový obvod, který má hodnotu BRAM s dostatečnou rezervou. Část BRAM bloků bude totiž využita i pro jiné použití, než je samotné uložení vzorků. A/D převodník vyžaduje pro danou konfiguraci alespoň 8 transceiverů. Vzhledem k těmto skutečnostem pak vyhoví všechny obvody v tabulce, kromě obvodu XC7K70T (nedostatečná BRAM). Všechny obvody mají taktéž dostatečný počet I/O pinů, například pro připojení externí paměti s dostatečnou šířkou sběrnice. Obvod XC7K325T splňuje všechny tyto požadavky při nejnižší ceně.

6.2.4 Vývojové desky s FPGA

Pozornost byla zaměřena opět na platformu Xilinx. Rozhodujícím faktorem byl především osazený obvod FPGA (viz předchozí kapitoly), velikost a typ externí paměti RAM a také fyzickým provedením konektoru (FMC, resp. FMC+) tak, aby bylo možné jednoduše připojit desku s A/D převodníkem. U výstupního konektoru je třeba zohlednit, jaké piny jsou na konektor vyvedeny – kritické pro naše použití je počet transceiverů, ne vždy jsou vyvedeny všechny transceivery. Periferie jako tlačítka, LED nebo displeje nebyly pro tuto aplikaci jakkoliv rozhodující.

Tabulka 6.4 Přehled vyhovujících vývojových desek firmy Xilinx - výběr

Označení	Rodina	RAM [Mb]		Počet transceiverů na FMC konektoru	Typ konektoru	Cena CZK
		Bloková	Ultra			
VCU118	Virtex US+	75,9	270	24x 28Gbps GTY	FMC+ HSPC, FMC HPC	153890
VCU108	Virtex US	60,8	0	10x GTH + 10xGTH	2 x FMC	131890
KCU105	Kintex US	21,1	0	8x GTX+1xGTX	FMC HPC + FMC LPC	65890
VC707	Virtex-7	37	0	8x GTX + 8xGTX	2 x FMC HPC	76890
Genesys 2	Kintex-7	16	0	10xGTX	1 x FMC HPC	28787
ML605	Virtex-6	14,9	0	8xGTX	2 x FMC HPC	43890
ZC706	Zynq 700	19,1	0	8xGTX,1xGTX	FMC HPC + FMC LPC	54890
10AX115S	Arria 10	-	0	15x	2 x FMC HPC	98890

S přihlédnutím k dostupné paměti (hlavně její rychlosti), poměrně modernímu osazenému FPGA obvodu a v neposlední řadě také ceně byla vybrána deska Genesys 2 s obvodem Kintex-7 od výrobce Digilent. Ta splňuje veškeré požadavky na tuto aplikaci. Jediným omezením je maximální rychlost GTX 10,3125 Gbps, pro FPGA s vyšším speed-gradem by bylo nutno zvolit desku VC707 či KCU105. Z důvodu tohoto omezení se tak bude nutno spokojit s hodnotou vzorkování do cca 5 GSa/s. Deska je blíže popsána v kapitole 6.3.3. Osazená je obvodem XC7K325T. Zajímavou volbou by byla i deska ZC706, a to kvůli dostupnému ARM procesoru na FPGA rodiny Zynq.

6.2.5 Paměťové obvody

Pro případ paměťových obvodů byla analyzována nabídka výrobce Micron, u kterého lze nalézt informaci o validaci obvodů pro obvody FPGA firmy Xilinx. Výrobce Micron nabízí (alespoň veřejně) pouze jediný obvod paměti typu HBM. Dále byly analyzovány obvody typu DDR2, DDR3 a DDR4. Tyto obvody jsou dostupné i ve formě modulů (např. SO-DIMM).

Tabulka 6.5 Vyhovující paměťové obvody - výběr

Označení	Kapacita [GB]	Frekvence [MHz]	Šířka sběrnice [bit]	Šířka pásma [GB/s]	Typ	Dostupný modul	Cena CZK
MT43A4G40200	2	neuvedeno	32/64	neuvedeno	HMC	NE	12672
MT40A256M16	0,5	1333	16	5	DDR4	ANO	198
MT40A512M16	1	1200	16	4,8	DDR4	ANO	330
MT41K512M16	1	800	16	3,2	DDR3L	ANO	484
MT41K256M16	0,5	933	16	3,7	DDR3L	ANO	176
MT52L512M64	4	933	64	15	LPDDR3	NE	1716
MT52L1G32	4	933	32	7,5	LPDDR3	NE	1716
MT53B256M32	1	1866	32	16	LPDDR4	NE	484
MT53B256M64	2	1866	64	30	LPDDR4	NE	1012

Pro potřebu rychlého ukládání by byla nejlepší volba paměť typu HMC, konkrétně obvod MT43A4G40200. Druhou alternativou je pak například obvod MT53B256M64. Ten poskytuje šířku sběrnice 64 bitů při frekvenci 1866 MHz. Žádný z obvodů není dostupný na modulu.

6.2.6 Paměťové moduly

Paměťové moduly už poskytují dostatečnou šířku sběrnice (vznikají sloučením několika diskretních paměťových obvodů na PCB), což zvyšuje reálnou propustnost paměti. Do přehledu byly zahrnuty i paměti SSD disků, které po sběrnici PCIe poskytují dnes poměrně velkou rychlost. Jedná se o standardní moduly, proto není třeba je zde uvádět a lze je nalézt pouze v rámci elektronické přílohy.

6.2.7 Kompletní řešení

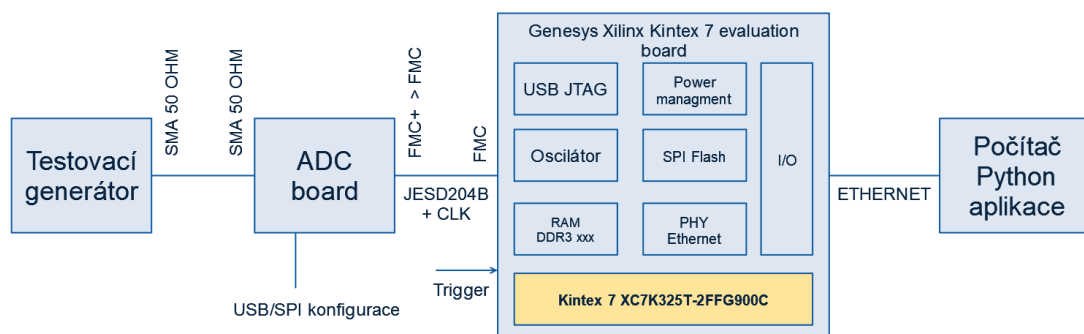
Kompletní řešení obsahují jak A/D převodník, obvod FPGA, tak i dostatečnou paměť. Bohužel však jednotlivá řešení nelze cenově porovnat – ceny u těchto řešení nejsou uvedeny. Buď se je nepodařilo zjistit ani po dotazu dodavateli nebo procházejí nabízené desky vývojovou revizí. Cena samotných obvodů FPGA však vždy přesahuje hodnotu 100 000 Kč. Kvůli obsáhlosti zde není přehled uveden (je ale uveden v rámci elektronické přílohy).

6.3 Popis hardwaru

Výsledný návrh je postaven na komerčně dostupných modulech – konkrétně vývojová deska Genesys 2 s FPGA XC7K325T (30 000,-Kč) a vývojové desky s ADC ADC12DJ3200 (60 000,-Kč). Výsledná cena je tedy přibližně 90 000,-Kč, v závislosti na dodavateli.

6.3.1 Blokové schéma

Na obrázku 6.1 je znázorněno blokové zapojení systému. Vstupní část práce může tvořit zesilovač, v rámci první fáze práce není tento blok ještě plně upřesněn. Lze však očekávat zesilovač s mezním kmitočtem 2,5 GHz (pro pokles o 3 dB) s DC vazbou a případně nastavitelným ziskem. Tento blok lze také na trhu najít jako již hotový modul. Druhým blokem je samotná vývojová deska s A/D převodníkem, do ní je přiveden měřený signál přes SMA konektor. Napájení desky je realizováno 5V zdrojem, který musí být dimenzován na proud až 3 A. Deska s A/D převodníkem je spojena s vývojovou FPGA deskou přes konektor FMC+, resp. s redukcí FMC. Konfigurace ADC desky zajišťuje převodník USB/SPI. Výsledná získaná data jsou poté zasílána do počítače pomocí rozhraní Ethernet.



Obrázek 6.1 Blokové schéma navrhovaného systému

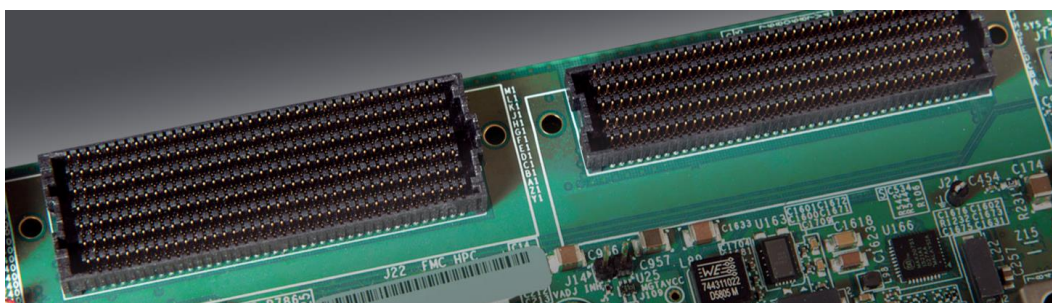
6.3.2 Standard VITA 57.1 a 57.4

Pro propojení jednotlivých bloků systému je použit standard FMC (FPGA Mezzanine Card). Standard byl vyvinut a schválen konsorciem dodavatelů FPGA a pracovní skupinou VITA 57. Jedná se o kompletní řešení elektromechanického rozhraní pro propojování jednotlivých modulů a desek určených pro použití s FPGA. Standard je uznáván výrobcí vývojových kitů a modulů pro FPGA. Standard definuje mapování jednotlivých pinů: single-ended signálů, diferenciální páry, transceivery, clock signály, JTAG, napájení, případně I2C nebo SPI.

FMC standard se dělí na dvě možné konfigurace konektoru:

- FMC (VITA 57.1), který se dělí na:
 - HPC, 400pinový, až 10 transceiverů
 - LPC, 160pinový
- FMC+ (VITA 57.4), který se dělí na:
 - HSPC, 560 pinový, až 24 transceiverů
 - HSPCe, 80pinový, až 8 transceiverů

Lze propojit i desky s různými konektory, avšak za cenu použití redukce. Fyzické provedení pro FMC a FMC+ je na obrázku 6.2. Použití redukce může pochopitelně způsobit problémy se signálovou integritou, pro použití s FMC konektorem je doporučována rychlost maximálně 10 Gb/s. Tato redukce je použita i v našem případě.



Obrázek 6.2 Konektor FMC + (vlevo) a FMC (vpravo), převzato z [48]

6.3.3 Vývojová deska s FPGA Genesys 2

Vývojová deska Genesys 2 (obrázek 6.3) od výrobce Digilent, je osazena FPGA obvodem XC7K325T-2FFG900C (tabulka 6.6) rodiny Kintex 7 firmy Xilinx. Tato deska je levnější varianta vývojové desky KC705 od Xilinx, ta však nemá paradoxně vyvedena dostatečný počet transceiverů na potřebný FMC konektor (některé jsou použity pro PCIe).

Tabulka 6.6 Základní vlastnosti FPGA XC7K325T [28]

Počet logických buněk	Počet DSP Slices	Počet CMT	Počet GTX transceiverů	BRAM
50950	840	10	16	16 Mbit
Počet I/O pinů	Počet I/O bank	Vnitřní CLK	Pouzdro	Cena CZK
460	10	450 MHz	BGA FFG900	35 464

Kromě samotného FPGA, vyřešeného power managementu a klasických periférií jako je displej (OLED), tlačítka a přepínače je deska vybavena obvodem fyzické vrstvy pro až 1Gbit Ethernet (obvod RTL8211E-VL), USB-JTAG programátorem, převodníkem USB-UART (obvod FT232R), 1 GB DDR2 RAM paměti (2 paměťové obvody Micron

MT41J256M16HA-107 po 16 bitech) s teoretickou šířkou pásma až 7,2 GB/s, SPI Flash paměti S25FL256S, dvěma oscilátory (200 MHz a 133 MHz). Přístup k pinům (a transceiverům) FPGA je umožněn přes FMC HPC konektor a několik dalších vyvedených pinů. Na desce jsou k dispozici také dva USB 2.0 porty. Napětí na vybraných napájecích větvích lze monitorovat pomocí obvodu INA219 (I2C sběrnice). Další výpis periférií je uveden v [28], pro naši aplikaci však nejsou podstatné.



Obrázek 6.3 Vývojová deska Genesys 2

6.3.4 ADC vývojová deska ADC12DJ3200EVM

Použitá vývojová deska (obrázek 6.4) od výrobce Texas Instruments je založená na AD převodníku ADC12DJ3200. Jeho parametry jsou shrnuty v tabulce 6.7 níže. Je vybavená obvodem LMX2582, který generuje clock signál pro vzorkování. Časová reference pro rozhraní JESD204B je založená na obvodech LMK04828 a LMX2582. Deska umožňuje přivést externí hodinový signál. Vstupní signál může být diferenciální i single-ended. Rozhraní JESD204B je vyvedeno na FMC+ HPC konektor (400pinový) s celkem až 16 páry (8 párů na kanál). Vstupní signál je diferenciální, s možností použití single-ended signálu o frekvenci 500 kHz až 9 GHz (balun). Výsledná deska je 16vrstvá s dielektrikem Rogers 4350B pro všechny vrstvy.

Tabulka 6.7 Základní vlastnosti ADC ADC12DJ3200 [29]

Rozlišení	Efektivní počet bitů	Počet kanálů	Vzorkovací rychlost	SFDR	Offset
12	8,5	2	3,2/6,4 GSa/s	67 dBFS	0,6 mV
DNL	INL	Rozhraní	Šířka pásma	Pouzdro	Cena CZK
0,3	2,5	JESD205B	8 GHz	144FBGA	55 000

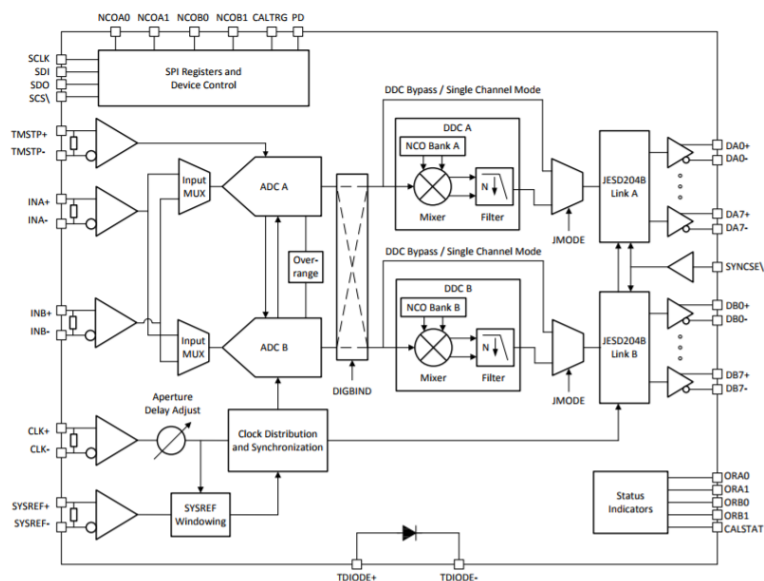
Samotný A/D převodník umožňuje vzorkovat vstupní signál od stejnosměrného až po frekvenci 10 GHz. Vnitřní struktura se skládá ze čtyř jader – dvě jádra pro každý kanál. Výsledné vzorkovací rychlosti je dosaženo prokládáním. V případě použití A/D

převodníků, jako jednonákanalového, je možno dosáhnout prokládáním ze všech čtyř jader rychlosti 6,4 GSa/s, čehož zde je využito. Konfigurace vnitřních registrů A/D převodníku probíhá přes SPI.



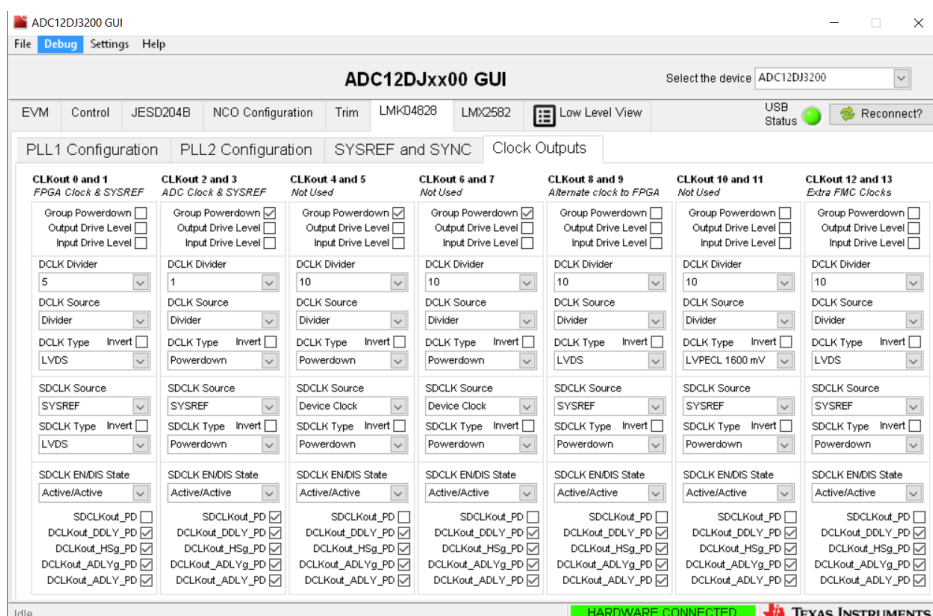
Obrázek 6.4 Vývojová deska ADC12DJ3200EVM

Na obrázku 6.5 je jeho vnitřní struktura. Vstupní signál je přiveden jako diferenciální, na vstupu může být aktivován terminační rezistor pro impedanční přizpůsobení. Multiplexer slouží pro přepínání kanálů na samotná jádra ADCA a ADCB. Výstup může být přiveden do digitálního downkonvertoru (DDS). Výsledná data jsou poté mapována (dle nastavení JMODE) na příslušné linky rozhraní JESD204B. Vstup TMSTP je vstup pro externí spouštěcí signál k časovému označení konkrétního vzorku (časové razítko). Vstupy NCOA0 až NCOAB1 slouží k řízení downkonvertoru. CALTRIG slouží pro kalibraci převodníku. PD slouží k vypnutí analogové části a serializéru v případě vysoké teploty. ORA0 až ORB1 jsou příznakové piny, které indikují překročení vstupního nastaveného rozsahu. Pin CALSTAT poskytuje příznak o stavu kalibrace, případně je nastaven jako příznak spuštění alarmu. TDIODE slouží k připojení externího teplotního senzoru (volitelně). Použitá architektura vychází z použití interpolace.



Obrázek 6.5 Vnitřní struktura A/D převodníku ADC12DJ3200

Poměrně omezující je problémová konfigurace obvodů na této desce. Rozhraní SPI není ani pro jeden obvod (ADC12DJ3200, LMK04828 a LMX2582) vyvedeno na FMC konektor ani na testovací body. Výrobce tak počítá s konfigurací desky skrze dodané GUI (obrázek 6.6). Na desce je použit převodník SPI/USB (výrobce FTDI). Je tedy nutné se spokojit s touto možností, případně se na SPI napojit ve vhodných bodech, od toho však bylo, vzhledem k provedení desky a její ceně, odstoupeno.



Obrázek 6.6 GUI pro konfigurace ADC (na obrázku nastavení výstupních hodin)

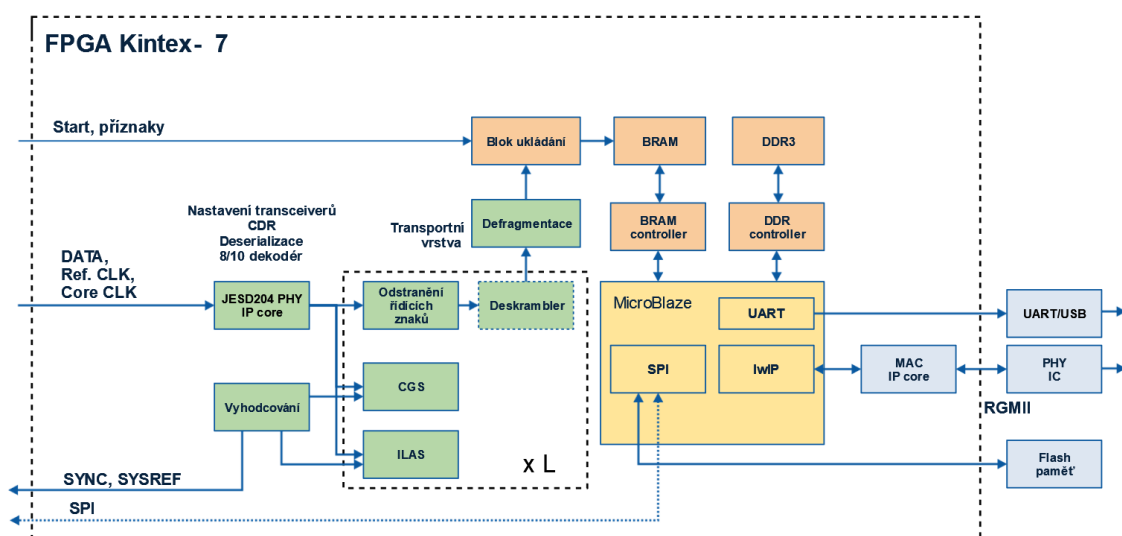
Z hlediska běžné aplikace je dnes využití desky i samotného ADC, v rámci země mimo USA, poměrně problematické. Export výrobku podléhá, z hlediska svých vlastností, regulacím. Vývoz těchto výrobků pro společnosti nebo jednotlivce mimo USA je možný pouze za určitých podmínek, především pak pro vzdělávací účely a civilní použití se zákazem exportu do třetích zemí.

7 FPGA DESIGN

V této kapitole je popsáno provedení implementace do obvodu FPGA. Návrh je postaven na několika IP jádrech, procesoru Microblaze a vlastních modulech VHDL, resp. C kódech pro Microblaze.

7.1 Koncept implementace

Na obrázku 7.1 je koncept implementace do obvodu FPGA. Pro přehlednost nejsou zakresleny všechny bloky (GPIO, PLL, ...). V základě lze implementaci rozdělit na tři hlavní bloky – JESD204B rozhraní, obsluhu paměti a samotný procesor Microblaze.



Obrázek 7.1 Implementace bloků do obvodu FPGA

Obsluha rozhraní JESD204B – provedení CGS, ILAS, nastavení a obsluha transceiverů, odstranění řídicích znaků z aplikačních dat, provedení defragmentace (odstranění rámcové a multirámcové struktury a seřazení vzorků ve správném pořadí). Výsledné vzorky jsou pak ukládány. Blok vyhodnocení řídí CGS a ILAS a kontroluje synchronizaci. Bloky lze rozdělit na vrstvu fyzickou (JESD204 PHY), vrstvu linkovou (IP od AD) a vrstvu transportní.

Práce s pamětí – ukládání vzorků získaných z ADC, poskytování dat procesoru skrze kontroler. Použitá BRAM paměť je dvouportová, jeden port bude využívat Microblaze skrze kontrolér, druhý bude využit pro ukládání (vyveden do HDL popisu).

Embedded část – procesor Microblaze zajišťuje konfiguraci ADC a vyčtení MAC adresy skrze SPI, poskytuje rozhraní UART (použité při ladění aplikace) a především slouží pro odesílání dat skrze Ethernet. Implementován je lwIP stack, ten realizuje TCP/IP stack, data jsou předávána jádru TEMAC, který zajišťuje vrstvu MAC. Skrze rozhraní RGMII jsou pak data předávána obvodu fyzické vrstvy.

Klíčová IP jádra:

- JESD204PHY
- JESD204B Rx AD
- Microblaze
- AXI TEMAC
- AXI QUAD
- AXI UART Lite
- BRAM, BRAM controller

JESD204PHY – blok pro obsluhu fyzické vrstvy JESD rozhraní. Především se jedná o konfiguraci transceiverů, nastavení přenosové rychlosti, počtu linek, typ ekvalizace, apod. Jádro je nelicencované. Částečně podobné je mj. jádro Tranceivers Wizard, který má v sobě i podporu pro vybraná rozhraní, vč. právě JESD204.

Microblaze – 32bitový soft procesor. Jedná se o procesor s redukovanou instrukční sadou. Na obvodu Kintex 7 (speedGrade 3) lze dosáhnout pracovní frekvence až 236 MHz. Nastavením lze ovlivnit výkon, případně zabrané prostředky FPGA (plocha, BRAM, DSP). Lze implementovat OS Linux. V práci je využit především pro implementaci lwIP stacku (zasílání dat do PC) a SPI komunikaci (konfigurace ADC) či I2C (případný vstupní zesilovač). V rámci této práce je jeho pracovní frekvence 100 MHz.

AXI Temac (Tri Mode ethernet MAC) – jádro pro obsluhu MAC vrstvy Ethernetu. U obvodů 6. generace se jednalo o hard jádro (na obvodu bylo několik těchto HW bloků), na které nebyla třeba licence. Od 7. generace je již provedeno jako soft jádro a je licencováno. Podporuje standardy 10, 100 i 1000 Mbit Ethernet.

AXI QUAD SPI – jádro pro SPI komunikaci, umožňuje práci v rozšířeném módu (Dual, Quad). V rámci této práce se používá mód Standard i QUAD. Rychlost je v rámci jádra nastavena parametrem „Clock ratio“. Počet obvodů slave je libovolný. Při použití Microblaze SDK automaticky importuje příslušné knihovny.

AXI UART Lite – jádro pro komunikaci skrze UART. 16 znakový buffer, nastavitelný počet datových bitů, počet stop bitů, volba parity a volba rychlosti. Opět importuje v rámci SDK příslušné knihovny.

BRAM Generator – generování blokové paměti na základě dané konfigurace.

BRAM controller – zpřístupnění BRAM, skrze sběrnici AXI, pro procesor.

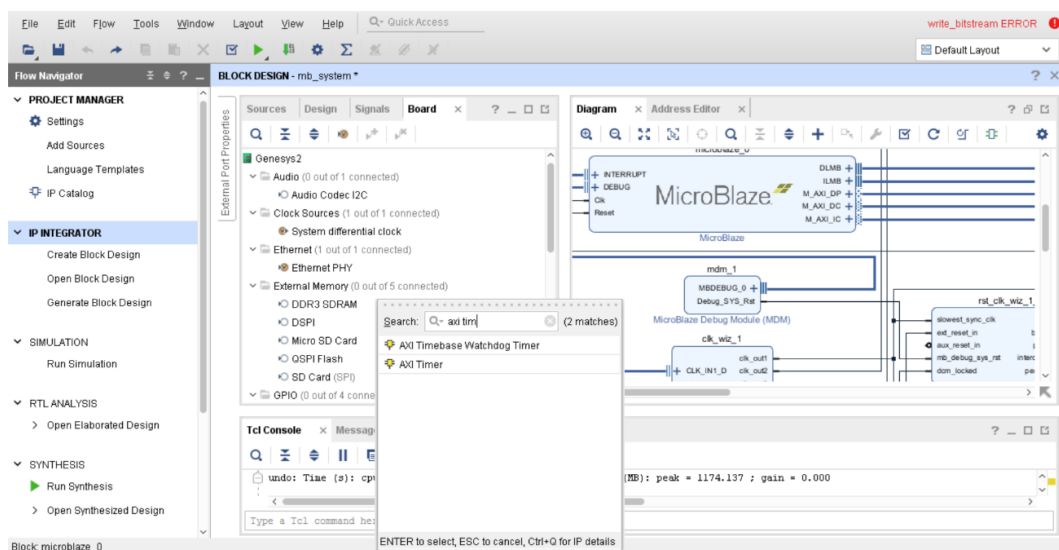
Další bloky slouží především jako bloky pomocné pro správnou funkci celého systému, jedná se o bloky pro přerušení, časovač, lokální paměť pro Microblaze, DMA řadič, nastavení PLL, apod. Často jsou doplňovány automaticky na základě nastavených parametrů jiných jader, resp. bloků.

7.2 Programování FPGA Xilinx

Veškerá práce s obvody FPGA Xilinx probíhá v případě této práce v prostředí Vivado 2017.3 (obrázek 7.2), resp. SDK 2017.3. Využito je revize VHDL-2008, která umožňuje například čtení výstupů, víceřádkové komentáře ve stylu jazyka C nebo zjednoduší práci s procesy, kde stačí v rámci sensitivity listu uvést klíčové slovo „all“, další rozšíření revize 2008 nejsou využity. Při použití revize 2008 je nutno dbát na to, aby všechny další nástroje použité v designu s tímto standardem byly kompatibilní, v opačném případě dochází k chybám.

Pro práci s použitým obvodem XC7K325 je třeba licence (vázano na MAC adresu, případně ID disku). V případě starších obvodů nebo při použití nástroje ISE se lze setkat s nástroji EDK a XPS. Kromě přímého popisu ve VHDL (či Verilog) je využit BlockDesign (mj. náhrada XPS), v němž lze integrovat patřičné bloky a následně vytvořit tzv. VHDL Wrapper, který lze již instanciovat nebo použít samostatně, v případě, že další popis FPGA není nutný. Při tvorbě lze využít nástroje Block automation, který automaticky doplňuje a připojuje potřebné periferie k danému bloku, případně Automation connecton, který automaticky bloky spojí a vytvoří externí porty. V případě, že je v projektu vybraná přímo vývojová deska a Vivado má potřebné informace, lze automaticky přidávat periferie desky (DDR, UART-USB, ...) a nástroj k nim automaticky vytvoří potřebné porty včetně XDC souboru. Kromě automatiky je nutné většinou návrh dodělat ručně. Automaticky vytvářený XDC soubor často při generování bitstreamu hlásí chybu, i toto je nutné kontrolovat a případně opravit. Případně lze použít vlastní XDC soubor, který je použit přednostně. Soubor typu XDC je náhrada souboru typu UCF.

Pro programování a debuggování procesoru Microblaze je pak využito prostředí SDK (Software Development Kit) postaveného na Eclipse. Podle exportovaného HW a zvoleného cílového obvodu automaticky importuje vhodné BSP (Board support Packages) a vstupně/výstupní porty. V případě úprav HW je třeba toto prostředí uzavřít, následně v nástroji Vivado znovu exportovat hardware a nástroj SDK znovu spustit. V opačném případě překladač často hlásí chyby.



Obrázek 7.2 Tvorba blokového designu v prostředí Vivado

7.3 Konfigurace ADC

Obsluha A/D převodníku se skládá ze dvou částí – konfigurace vnitřních registrů (a jejich čtení) a samotného přenosu dat. Obě dvě části jsou od sebe diametrálně odlišné. První část tvoří poměrně tradiční sběrnice SPI, což lze označit z hlediska dnešních embedded systémů za rutinní záležitost, druhou část lze pak označit za stěžejní část této samotné práce.

Pro dosažení korektní funkce A/D převodníku, zajištění bezchybného přenosu dat a dosažení nejlepších vlastností je nutné vhodně převodník nakonfigurovat a během měření zjišťovat jeho stav. Veškerá konfigurace je prováděna pomocí registrů. Veškeré informace lze najít v [30], kde je i uvedeno defaultní nastavení registrů, resp. nastavení registrů po resetu. Ne všechny registry je nutno nastavovat, resp. měnit jejich nastavení. Pro účely konfigurace je využito soft procesoru Microblaze. Samotný výpis všech registrů je značně obsáhlý, proto je níže zmíněno jen jejich základní rozdělení. Konfigurace se provádí přes rozhraní SPI. K tomuto účelu jsou vytvořeny příslušné symbolické názvy registrů jako direktivy pro překladač.

Standard SPI-3.0 (0x000 to 0x00F)

Pomocí registrů v této bance lze přistupovat k informacím ID a verzi daného obvodu (čipu), konfiguraci SPI rozhraní. Dále lze pomocí daných registrů detekovat AD převodník jako vysokorychlostní nebo schopnost používat rozhraní JESD204B. Současně je možno pomocí tohoto registru vyvolat reset.

User SPI Configuration (0x010 to 0x01F)

Tímto registrem lze nastavit způsob adresace, který bude použit při streamingu. Jedná se jen o nastavení jednoho bitu, zbylé bity jsou vyhrazeny pro pozdější použití.

Miscellaneous Analog Registers (0x020 to 0x047)

Tento registr slouží pro nastavení analogové části A/D převodníku. Jedná se o konfiguraci hodinových signálů, vstupních rozsahů nebo napěťové reference. Dále z něj lze číst informaci o časování.

Serializer Registers (0x048 to 0x05F)

Daný registr slouží pro nastavení prem-fáze sériového rozhraní. Nastavení je globální pro všechny spoje (16) rozhraní.

Calibration Registers (0x060 to 0x0FF)

Tento, poměrně rozsáhlý, registr (vzhledem k předchozím) slouží k veškerému nastavení kalibrace. Kromě povolení kalibrace jednotlivých částí A/D převodníků se zde definuje i samotné nastavení kalibrace. Prodleva mezi spuštěním A/D převodníků a spuštěním kalibrace, nastavení zisků jednotlivých kanálů, offsetu, terminace daných kanálů, nastavení vzorkování pro jednotlivá jádra, potlačení šumového pozadí, apod. Ne všechny registry je nutno pro standardní chod nastavovat (jsou nastaveny defaultně).

ADC Bank Registers (0x100 to 0x15F)

Nastavení časování jednotlivých bank. Rozděleno zvlášť pro nastavení fáze 0° a 90°.

LSB Control Registers (0x160 to 0x1FF)

Při nastavení tohoto registru je časová značka přenášena na pozici LSB. Tato funkce je podporována pouze tehdy, když není prováděna decimace.

JESD204B Registers (0x200 to 0x20F)

Slouží pro zapnutí/vypnutí JESD204B rozhraní a jeho nastavení (především JMODE). Nastavením je myšleno například počet rámců v multirámcí, nastavení synchronizace, kódování, scrambling. Slouží dále pro nastavení testování. Pro čtení slouží stavové registry.

Digital Down Converter Registers (0x210-0x2AF)

Obstarává nastavení digitálního down konvertoru. Pro účely této práce se nevyužívá.

SYSREF Calibration Registers (0x2B0 to 0x2BF)

Nastavení synchronizace a kalibrace.

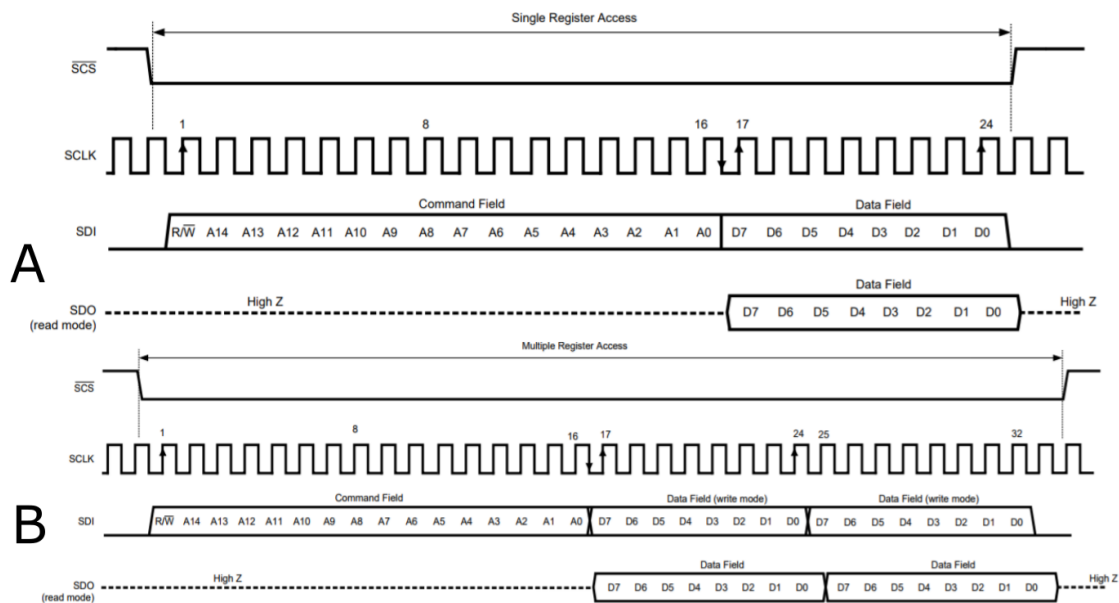
Alarm Registers (0x2C0 to 0x2C2)

Stavový bit v tomto registru je nastaven, pokud dojde k jakémukoliv problému v rámci převodníku.

Pro samotnou konfiguraci využívá převodník rozhraní SPI, konkrétně tedy piny: SCS, SCLK, SDI a SDO. SCS má funkce chip select – pro přístupu k registrům musí být tento pin v nízké úrovni, nutno je dodržet parametry setup/hold time vzhledem k hodinovému signálu SCLK. Signál SCLK je hodinový signál, není zespolu frekvenčně omezený, sériový přenos reaguje na nástupnou hranu tohoto signálu. SDI (seriál data in), tento signál přenáší data směrem od procesoru (Microblaze) k převodníku. Signálem SDO (Seriál Data Out) jsou přístupná data v případě čtení. Rámec přenosu má délku 24 bitů (obrázek 7.3 A) – první bit udává, zda se jedná o zápis nebo čtení, poté následuje 15bitová adresa daného registru a 8bitová data. V případě čtení se posledních 8 bitů ignoruje. Data jsou přenášena v pořadí od MSB po LSB. Zvláště pro prvotní konfiguraci je výhodné použít tzv. streaming mód (obrázek 7.3 B). Tento se nastavuje pomocí příslušného registru. V případě aktivace tohoto módu je adresa inkrementována automaticky a probíhá tak pouze zasílání dat pro zápis, v tom případě však musí probíhat postupně zápis do všech registrů, resp. od výchozí adresy.

Podle doporučení výrobce by se mělo na sériové rozhraní přistupovat až ve chvíli, kdy byla dokončena kalibrace převodníku. Stejně tak výrobce nedoporučuje provádět konfiguraci v průběhu samotného měření.

Pro SPI je využito jádro AXI Quad SPI (viz kapitola 7.1), v rámci SDK již importuje příslušné knihovny. Přenos je možný až čtyřnásobnou rychlostí, což je v případě konfigurace zbytečné a je využito módu standard. Délka transakce bude nastavena na 8 bitů (volitelně lze 8, 16 a 32 bitů). Rychlost přenosu je aktuálně nastavena na 400 kHz (nastaveno parametrem Clock ratio) a lze ji kdykoliv zvýšit.



Obrázek 7.3 Průběh SPI konfigurace, A – standardní mód, B – streaming mód, převzato z [46]

V rámci praktické realizace se však ukázalo, že deska nemá vyvedeno SPI rozhraní na FMC konektor a její konfigurace je tak založena pouze na dodaném GUI skrze převodník USB \leftrightarrow SPI. Vzhledem k ceně desky a jejímu provedení (16 vrstvá) by bylo riskantní se snažit desku vhodně upravit. Totéž platí i obvodech pro časování (LMK04828 a LMX2582). V rámci práce byla tedy pouze ověřena funkčnost komunikace SPI na Microblaze s obecným SPI zařízením (RFID čtečka). Dále je vytvořena funkce (avšak není z výše uvedených důvodů odladěná) pro konfiguraci registrů ADC. Pro obvody LMK04828 a LMX2582, zajišťující časování, by byl kód obdobný, avšak není jasné, zda budou přímo tyto obvody v budoucnu využity, klíčový aktuálně zůstává jen A/D převodník.

Z hlediska GUI je nutno vybrat požadovanou vzorkovací rychlost a nastavit patřičné parametry JESD204B rozhraní a dále děličky pro generování hodin (viz dále). Několik hodnot bylo nutno nastavit na úrovni registrů. V případě realizace vlastní desky pak postačí vhodně nakonfigurovat všechny tři obvody v rámci GUI a získat potřebný obsah registrů, což usnadní následnou vlastní konfiguraci. GUI následně umožňuje i konfiguraci obvodů LMK04828 a LMX2582. Použití GUI na druhé straně umožní snadno experimentálně optimalizovat nastavení A/D převodníku, pokud se jedná o nastavení parametrů JESD, případně módu je vhodné vždy provést reset transceiverů, aby byl znovu inicializován proces ekvalizace.

7.4 JESD204B rozhraní

7.4.1 Implementace fyzické vrstvy

Pro správnou funkci rozhraní JESD204B je využito, v závislosti na implementaci, až 5 druhů signálů:

- Linky vysokorychlostních transceiverů, u použitého FPGA se jedná o GTX transceivery (diferenciální). Linkou je myšleno v této práci anglické označení „Lane“.
- Referenční časový signál pro blok CDR v rámci transceiverů (diferenciální), označován jako „refclk“
- Časový signál pro JESD204PHY jádro a jeho logiku (diferenciální), označován jako „coreclk“
- Signál /SYNC (diferenciální nebo single-ended)
- Signál SYSREF (diferenciální)

Signály SYSREF a „coreclk“ nemusí být nutně využity vždy. SYSREF se nevyužívá u subclass 0, coreclk naopak nemusí být použit u GTX transceiverů. Označování signálu typu CLK se liší v závislosti na použitém jádře a jeho konfiguraci.

Samotné použité FPGA je vybaveno 16 GTX transceivery, pro daný speed-grade je rychlost jednoho GTX až 10,3125 Gbps. Na konektor FCM je vyvedeno však jen 8 transceiverů. Pro patřičnou konfiguraci ADC je nutno využít 8 nebo 16 transceiverů, z čehož vyplývá, že bude využito 8 transceiverů (mód ADC JMODE0). Celková datová propustnost činí 82,5 Gbps. Pro danou propustnost tak může být maximální vzorkovací rychlost maximálně 5 GSPS (datový tok 80 Gbps).

Rychlost vzorkování nelze volit libovolně pro použití plné propustnosti. Pro plnou vzorkovací rychlost by bylo nutné využít buď jinou vývojovou desku (vysoká cena) nebo využít vlastní konstrukci a využít všech 16 transceiverů. Kompromisem by bylo snížení rozlišení, což by vedlo ke snížení celkového datového toku, díky čemuž by bylo možné zvýšit vzorkovací rychlost. Využití většího počtu transceiverů by vedlo k rozložení datového toku na jednotlivé linky, což má ve finále i pozitivní vliv na signálovou integritu, požadavky na časování, apod.

V rámci základního nastavení transceiverů je nutno definovat rychlost (Linerate) a referenční hodinový signál, který využívá blok CDR (Reference clock, refclk). Patřičný Linerate může být vypočítán na základě vzorkovací rychlosti (při uvažování 8/10 b kódování a tail bitů), výrobce však již uvádí přepočtení konstanta R [Fbit/Fclk] (počet bitů na jedné lince za jeden takt hodinového signálu ADC). Platí:

$$Linerate = R \cdot DEVCLK, \quad (10)$$

kde $DEVCLK$ je hodinový signál převodníku.

Je však nutno si uvědomit, že hodinový signál ADC není roven výsledné vzorkovací rychlosti, hodnota $DEVCLK$ je poloviční, jelikož vzorkování probíhá na nástupnou i sestupnou hranu tohoto signálu. V dokumentaci k ADC je pro zvolený mód JMODE 0 uvedeno $R = 4 \text{ Fbit/Fclk}$, pro vzorkovací rychlost 5 GSa/s tedy konkrétně:

$$Linerate = 4 \cdot 2500 \text{ MHz} = 10 \text{ Gbps},$$

Referenční hodinový signál pro blok CDR by měl splňovat určitý poměr vzhledem k přenosové rychlosti. Pro použitou rychlost vzorkování je uváděna doporučená velikost [31], [32] jako $Linerate/20$, pro nižší rychlosti pak $Linerate/40$. Referenční časový signál je tak:

$$Refclk = \frac{Linerate}{20} = \frac{10}{20} = 500 \text{ MHz}, \quad (11)$$

Hodinový signál pro interní logiku GTX a navazující bloky je dán počtem bitů, které GTX zpracovává. Pro JESD204B je standardní režim práce s šířkou 40 bitů pro 1 linku, tj. 32 užitečných bitů. Tento signál bývá označován jako $coreclk$ (nebo obdobně) a měl by být roven hodnotě $Linerate/40$. Je získán buď přivedením tohoto signálu z jiného zdroje, případně je odvozen z referenčního clocku, kde výstup $coreclk_out$ může přímo budít $coreclk_in$, podle dokumentace se však tato možnost doporučuje jen u transceiverů GTH (novější rodiny UltraScale a UltraScale +). Další možností je použití MMCM z jiného zdroje hodinového signálu, nicméně je tato volba dosti riziková z hlediska jitteru a nejistoty fáze. $Coreclk_out$ je pak použit pro časování následujících bloků (například pro transportní vrstvu). Jelikož obvody na desce s ADC umožňují generovat několik nezávislých a vzájemně svázaných hodinových signálů, je využita možnost externího zdroje. Nicméně experimentálně bylo ověřeno, že funguje i buzení odvozeným signálem z výstupu $coreclk_out$. Hodnota $core_clk$ je tedy 250 MHz.

Hodnoty $refclk$ i $coreclk$ jsou odvozovány od hlavního časového signálu, který řídí vzorkování. Pro 5 GSPS je tedy $DEVCLK = 2\,500 \text{ MHz}$, pro $refclk$ je nutno nastavit děličku výstupu obvodu LMK04828 CLKOUT 0,1 na hodnotu 5, pro $coreclk$ pak děličku CLKOUT 12,13 na hodnotu 10. Hodnoty jsou shrnuty v tabulce 7.1.

Tabulka 7.1 Nastavené parametry fyzické vrstvy

Linerate [Gbps]	DCLK [MHz]	Ref. CLK [MHz]	Core. CLK [MHz]	DCLK DIV 0,1	DCLK DIV 12,13
10	2500	500	250	5	10

Referenční clock pro GTX je přiveden na vstup bufferu IBUFDS_GTE2. Vzniklý single-ended signál již nesmí využít globální buffer, jelikož ten je instanciován přímo v samotném JESD204PHY jádru. Tento bod je opět silně závislý na použitém jádru a jeho konfiguraci, rodině FPGA a typu transceiverů.

Signál /SYNC slouží pro zahájení CGS (během CGS je v logické úrovni 0, při normálním režimu zasílání dat v úrovni 1). Pro použitý ADC je možno přivést signál SYNC na vstup TMSTP jako diferenciální nebo na normální vstup SYNC jako single-ended. Při praktickém testování se více osvědčil diferenciální vstup TMSTP. Při použití single-ended někdy docházelo k problémům s CGS. Tento vstup má primární funkci pro „časové razítko“ vzorků. Aby byl TMSP použit i pro SYNC je nutno změnit hodnotu v registru TMSTP_CTRL. Z hlediska použití vývojové desky je pak nutno změnit pozici osazených nulových rezistorů, ty budou propojovat vstup TMSTP s příslušnými piny FMC

konektoru namísto SMA konektoru, se kterým (v základní verzi) nulové rezistory vstup TMSTP spojují.

Při praktické realizaci byla zjištěna informace o invertování datových linek DB0 až DB3 (zaměněny p a n vstupy). Tato informace byla do dokumentace desky doplněna až v lednové revizi dokumentace, ze schématu desky není tento fakt na první pohled zřejmý. Pro korektní funkci je nutno invertovat dané linky přijímače i v jádru JESD204PHY, kdy je nutné z jádra vyvést porty pro debuggování a řízení. Vstupy gt0_rxpolarity_in až gt3_rxpolarity_in je nutno nastavit na hodnotu '0' = bez inverze, vstupy gt4-gt7 pak na hodnotu '1' = s inverzí.

Dále je nutno nastavit vhodný typ ekvalizace. V jádru JESD204PHY je v určitých verzích Vivado, resp. IP jádra problém, kdy je (bez vyvedení debuggovacích a řídicích portů) nastavena první linka na ekvalizaci LPM a zbylé na ekvalizaci DFE, což způsobuje chyby při přenosu. S rozhraním JESD204B je kompatibilní ekvalizace LMP. Z toho důvodu je nutné nastavit vstupy gt0_rxlpmen_in až gt_rxlpmen_in na hodnotu '1' = LMP ekvalizace. Byť by měla ekvalizace DFE poskytovat lepší výsledky, není například vůbec vhodná pro testovací režim rampové funkce, který ADC nabízí. Dále lze vypnout vysílací linky. Ostatní porty je nutno nastavit dle dokumentace do defaultních hodnot tak, aby nebyla činnost jádra jakkoliv změněna. Příklad pro špatné nastavení ekvalizace je na obrázku 7.4.

Kromě jádra JESD204PHY je možné použít i jádro Transceivers wizard, kde lze v nastavení vybrat přímo protokol JESD204B. V jádru JESD204PHY je toto jádro vnitřně instanciováno.

Name	Value	633	634	635	636	637	638	639	640
> gt0_rsch...isk[3:0]	d			d			f		e
> gt0_rdata[31:0]	bcbca006	bc40f906	bc82f906	bcbca006		5cbcb7d	5cbcbcb		bef906f9
> gt1_rsch...isk[3:0]	a				a			c	
> gt1_rdata[31:0]	bca006f9	bcbcbcb	5cbcbcb	41bcbcb	bca006f9		bcbcbcb	bcbcbcb	bcbcbcb
> gt2_rsch...isk[3:0]	9	f	e	6	9		f		d
> gt2_rdata[31:0]	bcbcbcb	bcbcbcb	bcbcbcb	83bcbcb	bc40f906	bc82f906	bcbcbcb	bcbcbcb	bcbcbcb
> gt3_rsch...isk[3:0]	c			c				7	9
> gt3_rdata[31:0]	bcbcbcb	bcbcbcb	bcbcbcb	bcbcbcb	bcbcbcb	5cbcbcb	bcbcbcb	bcbcbcb	bcbcbcb
> gt4_rsch...isk[3:0]	f				f		e		f
> gt4_rdata[31:0]	bcbcbcb	bc40f906	bcbcbcb	bcbcbcb	bcbcbcb	bcbcbcb	bcbcbcb	fcbbcb	03bcbcb
> gt5_rsch...isk[3:0]	e		f	7			c		
> gt5_rdata[31:0]	bcbcbcb	5cbcbcb	bcbcbcb	bcbcbcb	bcbcbcb	bcbcbcb	bcbcbcb	bcbcbcb	bcbcbcb

Obrázek 7.4 Příklad zachycených dat CGS analyzátozem pro špatně nastavenou ekvalizaci

7.4.2 Implementace linkové vrstvy

Jelikož je rozhraní JESD204 poměrně nové a zatím méně rozšířené mezi běžnými uživateli, než běžná rozhraní, je jeho praktická implementace poměrně málo zdokumentovaná. Většina aplikačních poznámek předpokládá využití komerčně dostupného jádra. Cena komerčně dostupného IP jádra nabízeného firmou Xilinx je přibližně 180 000 CZK, v rámci licence se nabízí plná podpora výrobce. Z hlediska implementace je poměrně bohatý zdroj informací oficiální standard [21].

Nicméně, v [33] lze nalézt návod na implementaci JESD204B, vč. potřebných bloků napsaných v jazyce Verilog poskytnutých firmou Analog Devices. Popis bude třeba upravit pro konkrétní parametry rozhraní použitého v této aplikaci. Implementace

rozhraní se shoduje s obrázkem 7.1.

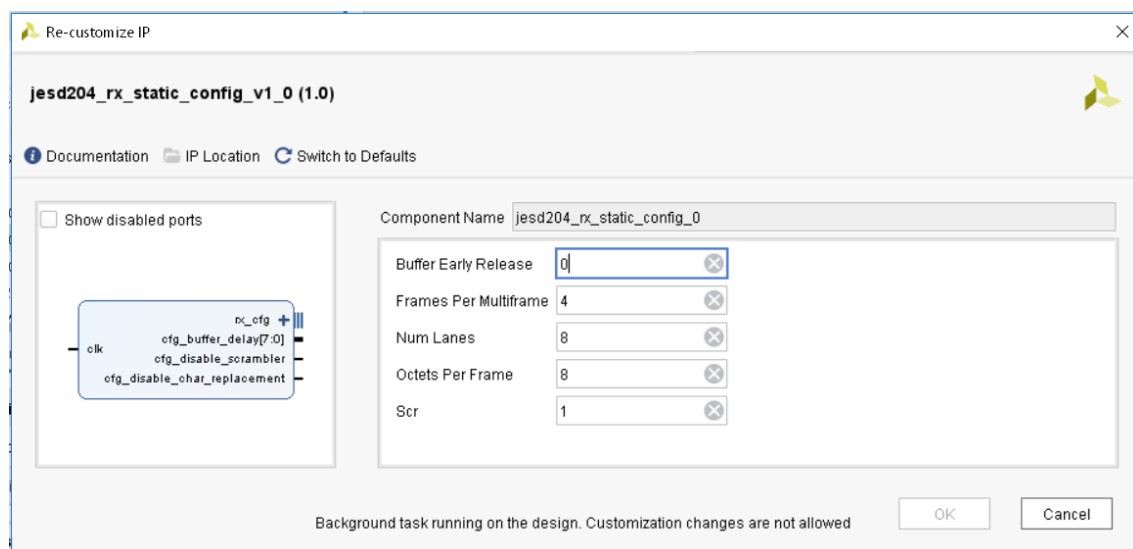
Oproti licencovanému jádru od firmy Xilinx má dané jádro jistá omezení, především nutnost využití scramblingu, v opačném případě nedochází k odstraňování řídicích znaků. Další omezení spočívají v maximálním počtu rámců v multirámcu, apod., nicméně tato limitace zde není důležitá, neboť alespoň pro použitý režim ADC jí nebude v žádném případě dosaženo.

Jádro je šířeno v rámci otevřených Verilog kódů, pro korektní funkci (mnoho parametrů) je nutno provést build jádra skrze dodané makefile soubory. Build jádra bylo nutné provést na OS Linux a se starší verzí Vivado, v opačném případě končil build chybou. I přesto bylo nutné vytvořené jádro otevřít a ručně doplnit dva chybějící Verilog moduly, jádro se jinak hlásí jako uzamčené a nelze jej konfigurovat nebo provést syntézu.

Přijímač se skládá ze dvou bloků – ze samotného „přijímače“, kde se konfiguruje pouze počet linek (L) a jádra pro konfiguraci. Přijímač lze konfigurovat buď skrze AXI nebo staticky (obrázek 7.5). Konfigurace skrze AXI je výhodná v případě, kde by se za chodu měnilo nastavení ADC, tím pádem i nastavení linkové vrstvy, apod. Aktuálně se využívá jádro pro statickou konfiguraci, výstupy jsou pak prakticky signály s konstantní hodnotou, připojené do samotného jádra přijímače.

V rámci tohoto IP jádra lze nakonfigurovat:

- **Buffer Early Release** – určuje, kdy budou data uvolněna z bufferu, pro deterministickou latenci je třeba hodnota 0, pro naše použití není toto nastavení důležité (ověřeno pro obě hodnoty nastavení)
- **Frames per Multiframe (K)** – počet rámců v multirámcu (pro použitý mód 4 až 32)
- **Num Lanes (L)** – počet linek (pro použitý mód L = 8)
- **Octet Per Frame (F)** – počet oktetů v rámci (8 pro použitý mód)
- **Scr** (1 = scrambler zapnutý, 0 = scrambler vypnutý)



Obrázek 7.5 Možnosti konfigurace JESD204B přijímače, použitá konfigurace

Nastavení je závislé na použitém módu A/D převodníku. Použitý mód je JMODE0. Parametry pro daný mód společně s několika dalšími módy jsou v tabulce 7.2.

Tabulka 7.2 Možné módy přenosu A/D převodníky (příklady)

Mód	Uživatelsky definované parametry		Odvozené parametry											
	JMODE	K	D	DES	Počet linek	N	CS	N'	L	M	F	S	R	
12bit, 1kanálový režim, 8 párů	0	3-32	1	1	2	12	0	12	4	4	8	5	4	
12bit, 1kanálový režim, 16 párů	1	3-32	1	1	2	12	0	12	8	8	8	5	2	
12bit, 2kanálový režim, 16 párů	3	3-32	1	0	2	12	0	12	8	8	8	5	2	
8bit, 2kanálový režim, 4 páry	6	18-32	1	0	2	8	0	8	2	1	1	2	5	

Tomuto nastavení JMODE tedy odpovídá 1kanálový mód převodník s 12bitovým rozlišením a rychlostí 6,4 GSa/s. Využito je 8 párů (8 transceiverů). Počet oktetů na rámec, resp. počet rámců na multirámec není pevně daný standardem, ale je určen danou (uživatelskou) konfigurací. Počet rámců v multirámcu je také nastavitelný příslušným registrem a pro zvolený režim lze hodnotu nastavit mezi 3-32 rámci (s krokem jednoho rámce). Způsob mapování vzorků na jednotlivé páry odpovídá pro JMODE = 0 (tabulka 7.3). Dále JMODE = 0 odpovídá:

- decimální faktor $D = 1$
- počet kanálů $DES = 1$
- počet linek = 2 (rozhraní je připojeno na výstupy obou kanálů)
- rozlišení převodníku $N = 12$
- ke vzorku nepřidává žádný kontrolní bit ($CS = 0$)
- celkový počet bitů je tedy $N' = 1$
- využity jsou čtyři páry na linku
- převodník obsahuje $M = 4$ jader (prokládání ze dvou jader na jeden kanál)
- rámec je složen z $F = 8$ oktetů, s celkem $S = 5$ vzorky v rámci
- během jednoho cyklu DEVCLK jsou vyslané 4 bity na jednom páru rozhraní

Pokud je vše korektně nastaveno, pak je výstupem tohoto jádra vektor šířky 256 bitů, data jsou zarovnána a řídicí symboly jsou odstraněny. Dalším výstupem je příznak validity dat, příznak startu/konce rámce a ostatní data pro monitorování (úspěšná CGS, fáze přenosu, apod.). Výstupem je samozřejmě také latence, informace o latenci však využita není. V následující vrstvě tedy stačí vzorky vhodně zpracovat (seřadit) a odstranit tail bity (mají hodnotu 0).

7.4.3 Implementace transportní vrstvy

Po fázi CGS a ILAS posílá AD převodník už samotné vzorky. Formátování vzorků a jejich mapování na jednotlivé linky je závislé na použitém módu A/D převodníku (nastavení v JESD204B Registers) – viz tabulka 7.2. Pro použitý mód JMODE0 je způsob mapování a formátování uveden v tabulce 7.3.

Tabulka 7.3 Mapování vzorků na jednotlivé páry pro JMODE = 0

Oktet	0		1		2		3		4		5		6		7	
Nibl	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
DA0	S0		S6		S16		S24		S32		T					
DA1	S2		S10		S18		S26		S34		T					
DA2	S4		S12		S20		S28		S36		T					
DA3	S6		S14		S22		S30		S38		T					
DB1	S1		S9		S17		S25		S33		T					
DB2	S3		S11		S19		S27		S35		T					
DB3	S5		S13		S21		S29		S37		T					
DB4	S7		S15		S23		S31		S39		T					

Jelikož transceiver zpracovává datovou šířku 40 bitů (32 bitů užitečných) je s každým taktem předáno transportní vrstvě 16 kompletních vzorků a 4 neúplné vzorky. Konkrétně dostáváme v prvním taktu 0. až 7. nibl, v dalším taktu pak 8. až 15. nibl. Pro ukládání, anebo veškeré další zpracování vzorků je žádoucí obdržet celistvý počet vzorků, a to ve vhodném pořadí. Transportní vrstva je koncipována tak, aby s každým taktem předala 20 vzorků (S0-19 a S20-S39). Pokud výstup přijímače přivedeme na dva za sebou zapojené registry, jsme schopni získat přístup k vzorkům z celého rámce a vzorky vhodně defragmentovat do vektoru šířky 240 bitů. Ty mohou být seřazeny v pořadí S0 až S19, resp. S20 až S39 nebo S19 až S0, resp. S39 až S20, pořadí už nehraje významnou roli, důležité je myslet na jejich pořadí při dalším zpracování.

Zpracování vzorků je pro každý „půlrámec“ jiné, pro tento účel je zavedena jednobitová proměnná, která s každým taktem mění svoji hodnotu, podle ní se pak mění způsob transformace dat na vzorky. Začátek rámce není pro správný začátek nutno znát, data začnou být validní se začátkem rámce. Výsledný blok může vypadat například následovně:

```
entity JESD204_transport_layer is
  Port ( rx_link_clk           : in STD_LOGIC;
         rx_link_clk_rst      : in STD_LOGIC;
         jesd204_rx_link_valid : in STD_LOGIC;
         jesd204_rx_link_data : in STD_LOGIC_VECTOR (255 downto 0);
         somf                  : in STD_LOGIC;

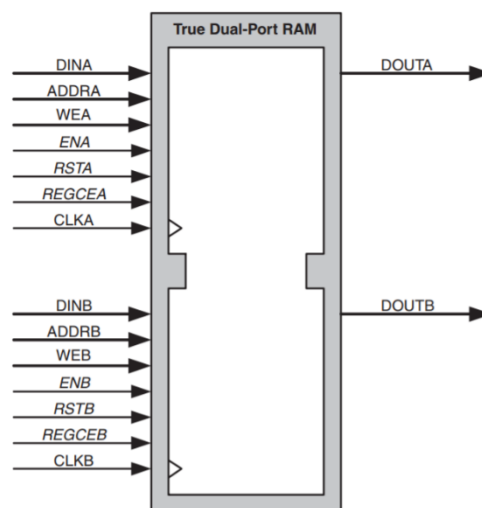
         rx_dataout            : out STD_LOGIC_VECTOR (239 downto 0);
         rx_validout           : out STD_LOGIC;
         rx_somfout            : out STD_LOGIC);
end JESD204_transport_layer;
```

7.5 Ukládání dat

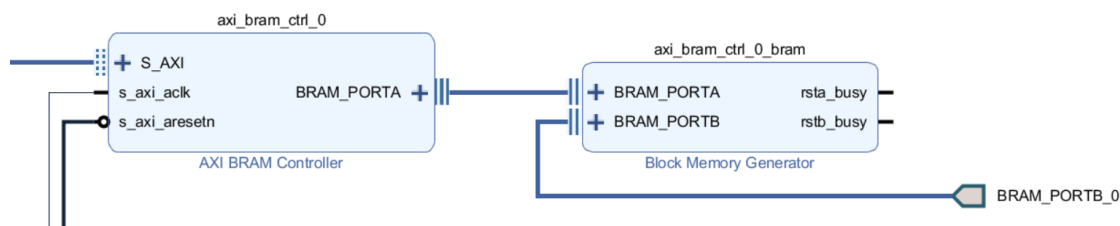
Limitem pro navrhovaný systém je omezená reálná rychlost zápisu externí paměti. Při požadované době záznamu $100\ \mu\text{s}$ s rozlišením 12 b a rychlostí vzorkování 6,4 GSa/s vychází potřebná minimální hodnota paměti cca 7 Mbit, resp. 640 tisíc 12bitových vzorků. V rámci použitého FPGA lze použít konfiguraci 36 Kb BRAM bloků (celkem 445) nebo 18 Kb bloků (celkem 890) [34], celková velikost BRAM je tak až 16 Mbit. Reálně je však část BRAM bloků zabrána například procesorem, nebo blokem pro MAC vrstvu Ethernetu. Při paměti s šířkou 256 bitů a hloubkou 32768 bude použito celkem 256 36Kb bloků, tj. BRAM vyhoví s jistou rezervou. Z použité šířky se efektivně využije 240 bitů (zbytek jsou tail bity), což je 20 vzorků, celkem je v této konfiguraci možné uložit 655 360 vzorků. Několik bloků BRAM zůstane stále volné pro případné rozšiřující bloky, případně zvětšení hloubky paměti.

BRAM (obrázek 7.6) může mít dva ekvivalentní a nezávislé porty pro zápis, resp. pro čtení s nezávislými CLK signály (True Dual port RAM). V aplikaci zde bude jeden port sloužit pro zápis, na druhém portu bude připojen AXI BRAM Controller. Pomocí AXI BRAM Controlleru je možno do paměti přistupovat skrze rozhraní AXI, což umožňuje mj. přístup do paměti z Microblaze procesoru. Z hlediska vygenerovaného block designu jako „black boxu“ bude přístupný pouze jeden port – pro zápis, toto je ilustrováno obrázkem 7.7.

Port pro zápis má nastavenou šířku 256 bitů – 240 bitů s transportní vrstvy je doplněno nulami. Takto přímý přístup do BRAM je adresován vždy celou šířkou, tj. adresa zde bude inkrementována o hodnotu 1. Druhý port a zároveň port AXI BRAM Controlleru má minimální šířku 64 bitů (minimální poměr 4:1). Zde probíhá adresování po jednotlivých bytech (v rámci Microblaze). Nutno je vhodně nastavit adresní prostor, tak aby odpovídal hloubce paměti.



Obrázek 7.6 Dvouportová paměť BRAM, převzato z [47]



Obrázek 7.7 Použití BRAM ve spojení s BRAM kontrolérem

Blok pro ukládání lze vytvořit například následovně (uveden spolu s popisem):

```
entity BRAM_trig is
  Port ( clk_link          : in STD_LOGIC;
         rst_in           : in STD_LOGIC;
         rx_data_in       : in STD_LOGIC_VECTOR (239 downto 0);
         rx_datavalid_in  : in STD_LOGIC;

         ext_trig         : in STD_LOGIC;

         BRAM_PORTB_addr  : out STD_LOGIC_VECTOR ( 31 downto 0 );
         BRAM_PORTB_clk   : out STD_LOGIC;
         BRAM_PORTB_din   : out STD_LOGIC_VECTOR ( 255 downto 0 );
         BRAM_PORTB_dout  : in STD_LOGIC_VECTOR ( 255 downto 0 );
         BRAM_PORTB_we    : out STD_LOGIC_VECTOR ( 31 downto 0 );
         BRAM_PORTB_en    : out STD_LOGIC;

         done             : out STD_LOGIC;
         error_out        : out STD_LOGIC;
         wait_out         : out STD_LOGIC;
         repeat           : in STD_LOGIC
  );
end BRAM_trig;
```

clk_link – clk signál, buzen z JESD204PHY bloku (rx_core_clk_out), což je LineRate/40

rx_data_in – data z bloku transportní vrstvy (20 x 12bitových vzorků)

rx_datavalid_in – příznak validity vstupních dat, nastaven do '1' pokud jsou data validní

ext_trig – vstupní signál, který spustí ukládání dat

BRAM_PORTB_XXX – porty samotné BRAM paměti (šířka Din a Dout 256 bitů)

Done – příznak dokončení ukládání, nastaven do '1' pokud bylo ukládání dokončeno (dosažení konenčné adresy)

Error_out – příznak chyby, nastaven do '1', pokud byla narušena validita, pokud byla data nevalidní, byť i během jednoho taktu clk, jsou uložená data vadná (narušená časová

7.6 Čtení a odesílání dat

Přesun dat do počítače je realizován skrze Ethernetové rozhraní. TCP/IP stack je řešen volně dostupným a poměrně uznávaným lwIP stackem (LightWeight), který je běžně používán i samotnými výrobci a je doporučováno i výrobcem Xilinx. Knihovna je šířena pod licencí BSD což mj. umožňuje její bezplatné využití. Kromě samotného protokolu TCP či UDP podporuje celou řadu dalších. Stack je implementován na procesoru Microblaze. Využito je tzv. RAW API, v tomto případě není třeba využít RTOS [35] (operační systém reálného času). Chod bez RTOS si lze dovolit, jelikož procesor nebude (kromě případné konfigurace ADC v další verzi) zastávat prakticky žádné kritické funkce.

Fyzickou vrstvu ethernetu zprostředkovává obvod Realtek RTL8211E-VL. Ten používá rozhraní RGMII pro data a MDIO rozhraní pro správu. Umožňuje použití 10, 100 i 1000Mbit ethernetu. MAC adresa obvodu je uložena ve Flash paměti v „one-programmable“ registru, ke kterému lze přistoupit přes SPI. Vrstva MAC (Media Access control) je realizována jako soft jádro na FPGA.

Jak bylo zmíněno, data nejsou odesílána v reálném čase. Je možno je přenést do počítače po ukončení měření. Data jsou uložena v paměti BRAM, procesor přistupuje do této paměti přes patřičný kontrolér a v podstatě tato data přenáší na základě dostupného místa do odesílacího bufferu lwIP stacku.

Čtení dat z paměti a jejich odesílání lze tedy rozdělit na několik kroků:

1. Vyčtení jedinečné MAC adresy z OTP registru ve Flash paměti skrze SPI
2. Případná inicializace DHCP serveru
3. Inicializace síťového rozhraní, definice IP, portu, masky
4. Čtení dat z BRAM, plnění bufferu a odesílání při dosažení požadovaného počtu vzorků v paketu

Jedinečná MAC adresa je uložena v externí Flash paměti v OTP registru na adrese 0x20, flash paměť využívá SPI rozhraní. Toto SPI je v designu samostatná komponenta (kvůli zapojení nelze využít SPI určenou pro ADC). Pro čtení slouží funkce `SpiReadOTP(&FlashSpi, MAC_OTP_ADDR, (u8*)&mac, sizeof(mac))`, ta vrací MAC adresu jako strukturu `char` a je nutno ji přetypovat. Parametrem je adresa registru, SPI komponenta a cílová proměnná.

Ve funkci `eth_init()` je vyčtena MAC adresa (viz předchozí bod) a jsou nastaveny patřičné IP adresy a port. V rámci aplikace lze volitelně využít získání IP adresy desky skrze DHCP, v tomto případě se pokouší rozhraní získat tuto IP adresu, po vypršení timeoutu je provedena kontrola, zda je adresa nenulová, pokud je nulová (IP adresa nebyla získána) je nastavena adresa defaultní. Funkci DHCP lze samozřejmě „vypnout“, kód pro DHCP je v rámci podmíněného překladu. Dále je inicializován samotný lwIP stack a je přiřazeno rozhraní pro obsluhu MAC vrstvy (soft IP jádro v rámci FGPA):

```
xemac_add(udp_netif,      &ipaddr,      &netmask,&gw,      (unsigned char
*)&mac, PLATFORM_EMAC_BASEADDR))
```

Konečně je pak nastaveno vytvořené rozhraní jako defaultní a je aktivováno:

```
netif_set_default(udp_netif);
```

```
netif_set_up(udp_netif);
```

Čtení dat z BRAM lze realizovat pro 32 bitů jako:

```
data= XIo_In32(XPAR_AXI_BRAM_CTRL_0_S_AXI_BASEADDR+4*ptr);  
resp. pro 8 bitů:
```

```
byte1=XIo_In8(XPAR_AXI_BRAM_CTRL_0_S_AXI_BASEADDR+offset+ptr++);
```

Tento přístup ke čtení je bytově orientován, tj. například při použití XIo_In32() je nutno pointer inkrementovat o hodnotu 4, nikoliv o hodnotu 1. Začátek adresního prostoru BRAM je označen jako XPAR_AXI_BRAM_CTRL_0_S_AXI_BASEADDR.

Pomocí výše uvedených funkcí jsou data čtená a ukládána do pole, při dosažení daného počtu prvků je pole předáno do odesílacího bufferu k odeslání:

```
memcpy (p->payload, buffer_send, sizeof(buffer_send));
```

Funkce `udp_sendto(pcb, p,&pc_ipaddr,port);` pak data předává k odeslání, dále je nutno volat funkci (mapování k MAC vrstvě):

```
xemacif_input(udp_netif);
```

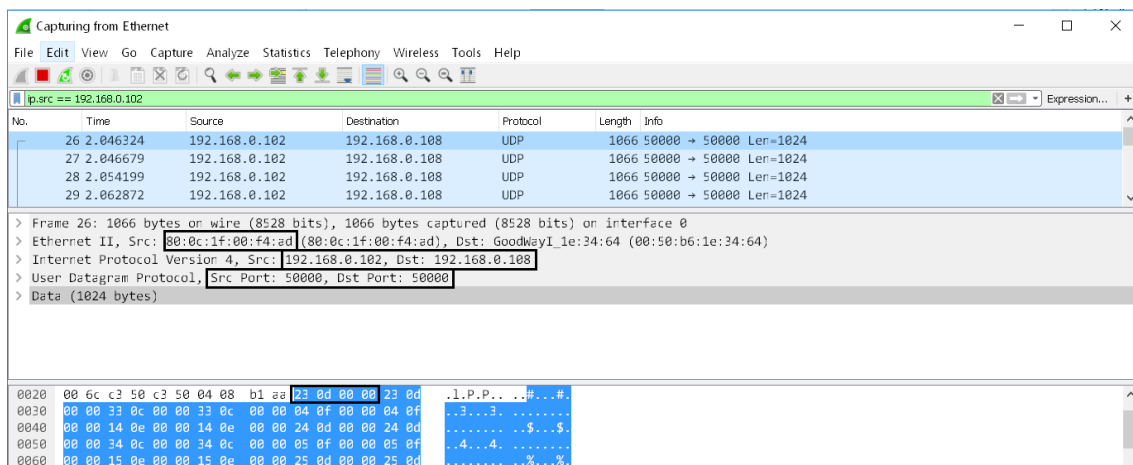
Při odesílání byla zvolena možnost, kdy jsou vzorky odesílány jako 32bitová čísla. V tomto případě jsou jednotlivé vzorky plně identifikovatelné v přenášených datech, ulehčuje to následné zpracování přijatých dat, ale především je pak příjem prakticky nezávislý na rozlišení A/D převodníku, až do rozlišení 32 bitů pak stačí úprava na straně vysílače, nikoliv na obou stranách. Nevýhodou této možnosti je však větší redundance pro použitý převodník s 12bitovým rozlišením. Tuto možnost je tedy nutno pečlivě zvážit, při odesílání dat v reálném čase s rychlým ADC (kde by rychlost odesílání byla vzhledem k rychlosti vzorkování dostačující) by toto možné být nemuselo.

V rámci ladění samotné implementace lwIP stacku byla skriptem v Matlabu vygenerovaná data (několik period funkce sinus v abs. hodnotě), která inicializovaly obsah BRAM paměti. Celkem bylo vygenerováno 200 tisíc 32bitových vzorků, které byly odeslány a porovnány se samotnými inicializačními daty ze souboru COE (*.coe). Porovnáním bylo zjištěno, že vzorky byly přijaty všechny a beze změny. Data lze pozorovat na obrázku 8.1 v následující kapitole.

V rámci ladění byla prvotní analýza prováděna programem Wireshark, na obrázku 7.10 níže pak je zobrazen detail jednoho z přijatých paketů. Pro ilustraci jsou použita získaná data z testu transportní vrstvy ADC, což se chová jako běžné vzorky. V programu lze například ověřit:

- IP adresa zdroje: 192.168.0.102 (nastavená v rámci Microblaze)
- IP adresa cíle: 192.168.0.108 (IP adresa přiřazená externí síťové kartě počítače)
- MAC adresa desky z Flash: 80:0c:1f:00:f4:ad
- TCP port zdroje, cíle: 50 000
- Protokol: UDP

V datech lze přímo identifikovat zasílané vzorky (zde jako Little Endian) jako 32 bitové hodnoty, konkrétně data 0xd23,0xd23, 0xc33,0xc33, 0xf04, 0xf04, apod.



Obrázek 7.10 Zachycené pakety programem Wireshark (data testu transportní vrstvy)

8 PŘÍJEM DAT

Pro příjem dat (vzorků) a jejich následný zápis je využito skriptovacího jazyka Python 3.x. Konkrétně pro obsluhu ethernetového rozhraní se jedná o knihovnu Socket [36]. Skript pak pro korektní funkci vyžaduje na daném počítači tzv. interpret, případně je možno vytvořit i spustitelný *.exe skript, nejedná se však o klasickou kompilaci jako v případě aplikací z jazyků C++ nebo C#, výsledný *.exe file je poněkud větší. Vzhledem k rychlosti vzorkování nejsou data odesílána v reálném čase, ale až po skončení měření. Vzhledem k rychlosti to pak umožňuje přímou separaci dat a zápis do CSV souboru, v opačném případě by bylo nutno využít například binární soubor.

Samotná funkce skriptu je poměrně jednoduchá. Na začátku je inicializováno samotné síťové rozhraní, jedná se především o:

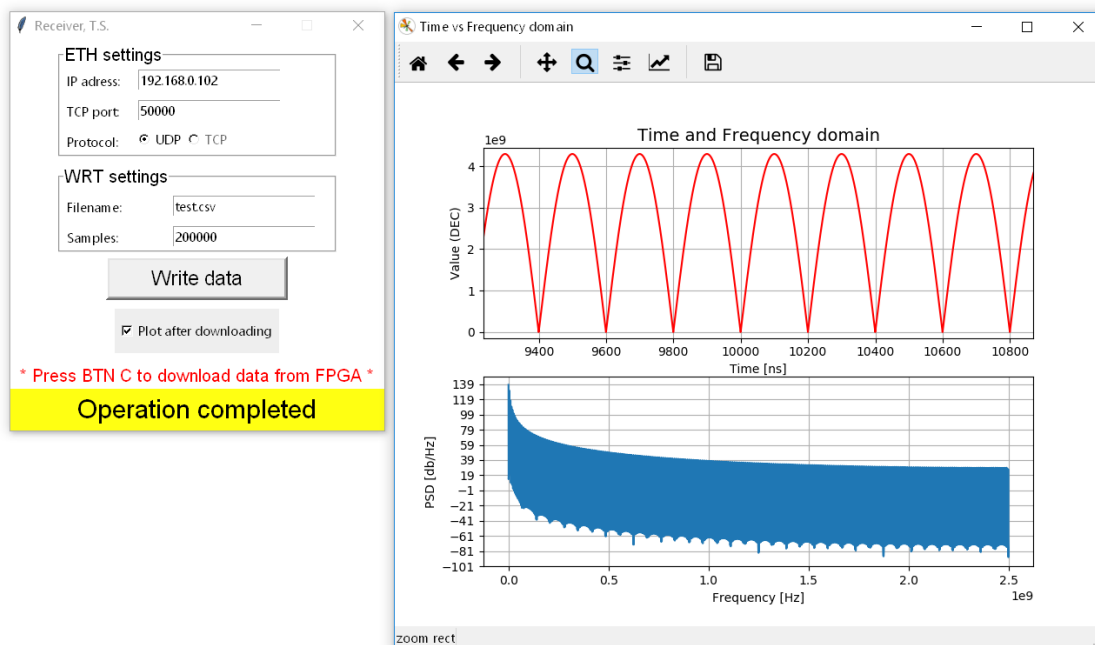
- Zdrojová IP adresa (host) – IPv4 adresa desky s FPGA, v případě, že je využita možnost získání IP adresy skrze DHCP server musí být tato IP adresa nastavena jako 0.0.0.0 (případně nastavena dle aktuální přidělené adresy)
- Číslo portu (port) – nastaví patřičný síťový port, v aktuální pracovní verzi skriptu je využit port s číslem 50 000 (oblast dynamických a soukromých portů)
- Timeout – nastaví timeout pro vytvořený socket v sekundách, mimo nastavenou hodnotu může být nulový nebo defaultní (nastavený systémem). Jelikož jsou data z FPGA odesílána kontinuálně, je zde využito timeoutu k ukončení příjmu dat. Druhou možností ukončení by bylo odeslání známé sekvence, apod.00000
- Parametr socket.SO_REUSEADDR ve funkci setsockopt, tímto parametrem je při znovuspuštění skriptu využit dříve vytvořený socket (je-li stále platný), bez tohoto parametru většinou nastane chyba, pokud je skript spuštěn hned po předchozím příjmu.

V další části je otevřen (vytvořen) patřičný CSV souboru s metodou with. Z přijímacího bufferu jsou čtena přijatá data, separována na jednotlivé vzorky a doplněna o časovou informaci. Data jsou z bufferu čtena jako bytearray, kde je nutno udělat konverzi na výsledné hodnoty vzorků. Po vypršení timeoutu (již nejsou žádná další data přijímána) je socket uzavřen, automaticky se uzavírá i CSV soubor (metoda with).

Pro ovládání bylo vytvořeno základní GUI založené na knihovně TkInter [37] a vykreslení pak na knihovně Matplotlib [38]. Vytvořené GUI je společně s příkladem vykreslení průběhu na obrázku 8.1. Veškeré zadávané údaje jsou předvyplněné a lze použít přímo je. V opačném případě je nutno zadat IP adresu desky a použitý port, toto nastavení musí souhlasit s konfigurací použitou v rámci konfigurace lwIP. Pro případné využití TCP portu v budoucnu počítá GUI i s touto možností (prozatím zakázáno). Dále je zadán název souboru, požadovaný počet vzorků k uložení, případně lze zaškrtnout volbu, díky které bude průběh po ukončení příjmu vzorků automaticky vykreslen v novém okně. Po aktivování příjmu lze pak data stiskem tlačítka na modulu FPGA odeslat.

Na obrázku 8.1 lze vidět příjem dat, kterými byla paměť inicializována, jedná se o vzorky funkce sinus v absolutní hodnotě. Těmito daty bylo testováno zasílání dat skrze Ethernet a jejich příjem – tato část funguje spolehlivě. Jediný, ale podstatný rozdíl je ten, že při tomto testu měla paměť šířku 32 bitů, nikoliv 256 bitů jako při zamýšleném použití.

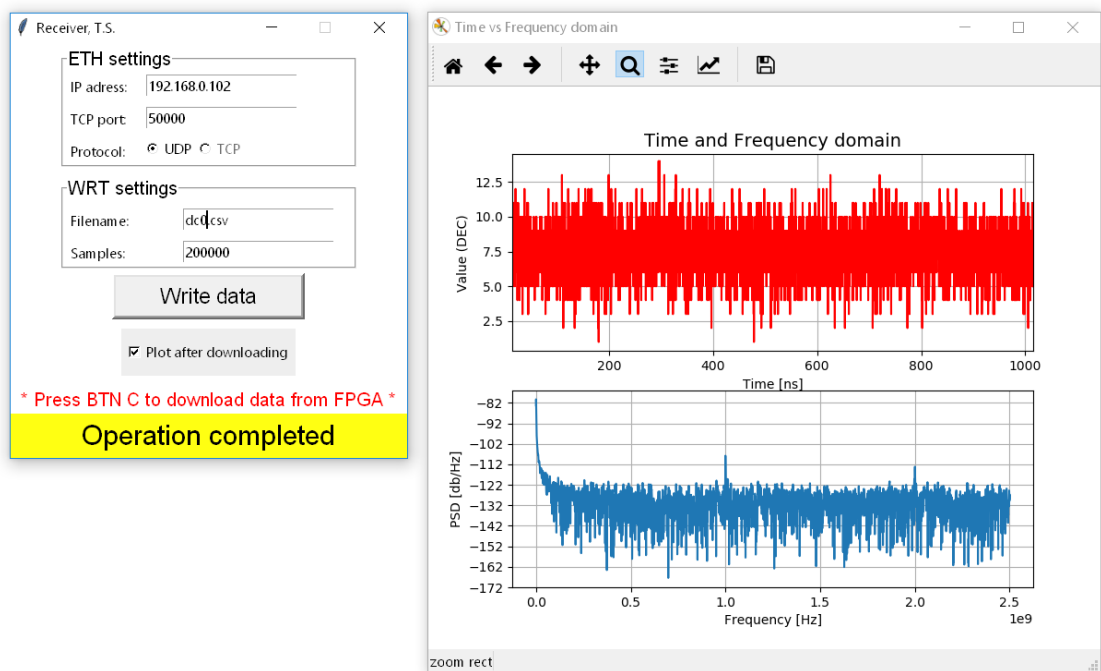
Právě tento rozdíl bude pravděpodobně příčinou problému, proč nefunguje korektně zasílání dat v běžném režimu, nicméně smyšlené použití možné je, avšak je nutno vyřešit, zatím neznámou, příčinu problému. Pravděpodobně se jedná o problém při čtení – buď problém s cache v rámci Microblaze nebo špatná konfigurace BRAM Controlleru, případně portu paměti k němu připojenému.



Obrázek 8.1 Příjem testovacích dat

Data z příjmů vzorků s ADC bez vstupního signálu jsou na obrázku 8.2, v tomto případě lze pozorovat jen šum, rušení, případně určitá rezidua vzniklá funkcí ADC. V tomto případě data působí vykreslena korektně, nicméně se zde jen neuplatní problém z předchozího bohu.

Aby mohla být znázorněna i data z dynamických měření, a jelikož se z důvodu nedostatku času nepovedlo vyřešit problém se čtením z paměti bylo využito provizorního řešení. V tomto případě jsou data z tohoto typu měření získána skrze integrovaný analyzátor, data jsou uložena do souboru *.CSV a následně zpracována programem Matlab do formy vhodné pro vizualizaci. Při této možnosti je nutno se spokojit s cca 20 tisíci vzorky, navýšením počtu vzorků v analyzátoru případně i 40 tisíc. Pro měřené hodnoty frekvence je však 20 tisíc dostatečných. Obrázky v následující části tedy nepochází z vytvořené aplikace pro příjem, ale z prostředí Matlab, data jsou však reálná a získána z ADC prostřednictvím FPGA. Nicméně řešení umožňuje v této fázi demonstrovat funkčnost ADC a korektního zasílání vzorků prostřednictvím JESD204B.



Obrázek 8.2 Přijatá reálná data z ADC

Name	Value
jescd204_0_rx_i/rx_sync	0
jescd204_0_rx_i/rx_tvalid	0
jescd204_transp...ataout[239:0]	0000000000000C
u_ila_0_gt_rcharisk[31:0]	00000000
u_ila_0_gt_rxdta[255:0]	4a4a4a4ab5b5b5f
GT7_data	4a4a4a4a
GT6_data	b5b5b5b5
GT5_data	b5b5b5b5
GT4_data	b5b5b5b5
GT3_data	4a4a4a4a

9.1.3 Zasílání dat rampové funkce

[illegible]

62

9.1.4 Zasílání dat testovací transportní funkce

V rámci tohoto testu ADC vysílá předem známý formátovaný obsah, který odpovídá několika módům přenosu, vč. námi použitého JMODE0. Perioda data je 8 oktetů, tj. po 8 oktetech (1 rámci) se datová sekvence začíná opakovat. V tomto režimu již bohužel selhávalo zasílání dat. V příloze lze najít srovnání vzorků získaných na základě transportní vrstvy s očekávanými hodnotami. Zachycení funkční transportní vrstvy je na obrázku 9.4.

Z hlediska praktických zkušeností s laděním transportní vrstvy by se jevílo vhodnější, kdyby ADC uměl posílat data rampové funkce odpovídající režimu ADC (tj. formátované 3niblová slova). Testovací sekvence transportní vrstvu má periodu jednoho rámce, což se při ladění nejevilo jako úplně vhodné.

Name	Value	191	192	193	194	195	196	197	198	199
> jesd204_rx_0_i_rl_state[1:0]	3						3			
↳ jesd204_rx_0_0rx_valid	1									
> jesd204_rx_0_i_intable[31:0]	00000000						00000000			
↳ jesd204_rx_0_n_char_align	0									
> jesd204_rx_0_n_dispers[31:0]	00000000						00000000			
> jesd204_rx_0_data[255:0]	c09a00a30b5f	6c1e9d	3e75aa712b7d	980cb6224d4d	e910f01f262d	a0c6457ea56d	c09a00a30b5d	974b342c14d	6f11b29fd5ad	d8d8517b594d
> jesd204_rx_0_n_charisk[31:0]	00000000									
↳ inter_lmfc_edge	0						00000000			
↳ jesd204_rx_0_0sync	1									
> jesd204_trans_er_0UQ[239:0]	d25d25c35c3f	f00f00	d25d25c35c3d	f00f00e11e1d	d25d25c35c3d	f00f00e11e1d	d25d25c35c3d	f00f00e11e1d	d25d25c35c3d	f00f00e11e1d
> jesd204_trans_data[255:0]	50c334c35d0;	c3321d	50c334c35d0d	c3321cec34d2d	50c334c35d0d	c3321cec34d2d	50c334c35d0d	c3321cec34d2d	50c334c35d0d	c3321cec34d2d
> u_ila_0_rx_eof[3:3]	1	0	0	0	1	0	1	0	1	0

Obrázek 9.4 Zachycená data pro mód testu transportní vrstvy

9.1.5 Opakované zasílání ILAS a analýza konfiguračních dat z ILAS

V tomto případě převodník na všech linkách opakovaně zasílá data inicializační sekvence JESD204B, ty obsahují povinně konfigurační data rozhraní, zbylá data pak mohou tvořit například data rampové funkce – jak bylo napsáno v kapitole 4.3. Aby se vyloučila nekorektní konfigurace ADC, tak byla konfigurační data z ILAS analyzována. Samotná data jsou v ILAS oddělena řídicím znakem K28.4 (0x9C). Data ILAS jsou zachycena na obrázku 9.5.

[illegible]

Obrázek 9.5 Zachycená ILAS pomocí analyzátoru (Chipscope)

Seskupením dat z fyzické vrstvy v rámci niekoľika taktů lze v oblasti konfiguračních dat ILAS obdržet:

.... 7C 1C 9C 01 00 01 03 07 03 03 0b 2b 24 00 00 00 2e 30 31 32, neboli:

0x7C: ohraničuje konec předešlého rámce ILAS

0x1C: začátek rámce

0x9C: začátek konfiguračních dat v ILAS (vyskytuje se pouze v ILAS a pouze jednou)

0x01-2e – konfigurační data

0x30-... - následující data rampové funkce, rozepsaná v tabulce 8.1

Tabulka 9.1 Analyzovaný obsah ILAS

Oktet	M S B							L S B	M S B								L S B
0	DID								0	0	0	0	0	0	0	0	1
1	ADJCNT					BID			0	0	0	0	0	0	0	0	0
2	X	ADJDIR	PHADJ	LID					0	0	0	0	0	0	0	0	1
3	SCR	X	X	L					0	0	0	0	0	0	0	1	1
4	F								0	0	0	0	0	1	1	1	
5	X	X	X	K					0	0	0	0	0	0	0	1	1
6	M								0	0	0	0	0	0	0	1	1
7	CS		X	N					0	0	0	0	1	0	1	1	1
8	SBCLS			N'					0	0	1	0	1	0	1	1	1
9	JESD			S					0	0	1	0	0	1	0	0	0
...

Konfigurační sekvence má celkem 13 oktetů, nicméně nejpodstatnější oktety, z hlediska konfigurace JESD204B jsou oktety 3 až 9. Většinu dat je nutno dekodovat jako binární hodnotu zmenšenou o 1. Na základě dat v tabulce 8.1 dekodovat:

- SCR = 0 (scrambler vypnutý)
- L-1=3 => L=4 (počet linek pro jeden kanál)
- F-1 = 7 => F= 8 (8 oktetů)
- K-1=3 => K = 4 (4 rámce v multirámci)
- M - 1 = 3 => M = 4 (počet jader)
- CS = 0 (0 kontrolních bitů)
- N-1 = 11=>N=12 (12bitové rozlišení)
- Subclass 1, N' = 12 (12 bitové rozlišení po započtení kontrolních bitů)
- Revize JESD204B,
- S-1 = 4 => S=5 (5 vzorků v rámci, celkem 20 ze 4 jader)

Tato konfigurace se shoduje s očekávanou a nastavenou konfigurací na straně přijímače (FPGA).

9.1.6 Diskuze

Jelikož přenos rampové funkce, ILAS a řídicích znaků probíhal po fyzické vrstvě korektně, bylo možné konstatovat, že se nejednalo o problémy s časováním nebo signálovou integritou. Stejně tak konfiguraci JESD204B na straně ADC a FPGA lze, díky shodě konfigurace v ILAS považovat za správnou.

Problém nebyl způsoben ani možnou špatnou lokací transceiverů (špatné mapování transceiverů k linkám ADC, apod.), v tomto případě by nebylo dodrženo pořadí celistvých vzorků, ne tedy na úrovni bitů. Vzhledem k tomu, že testovací sekvence byly zachyceny na všech transceiverech není problém způsoben ani připojením nevyužitých transceiverů nebo naopak nepřipojením potřebných transceiverů.

Vývojová deska by měla být napájena zdrojem dimenzovaný na proud až 3 A, špatně dimenzovaný zdroj může podle výrobce způsobovat jak chyby při přenosu, problémy s časováním (jak s časováním JESD204B, tak vzorkováním). Při provozu ADC a dalších obvodů je pokles napětí napájecího zdroje minimální a zvlnění dosahuje max. 10 mV, což je navíc zvlnění před regulátorem na vývojové desce. Tomuto problému by však stejně neodpovídaly vybrané funkční sekvence.

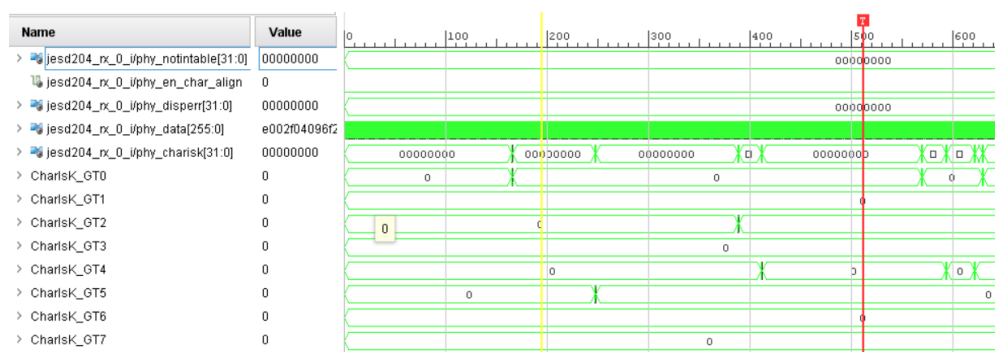
V rámci další experimentální práce při pokusech o korektní funkci ADC byla vyzkoušena i ekvalizace DFE (bez výsledku), byl vyzkoušen provoz pro Refclk odpovídající hodnotě Linerate/40 (namísto Linerate/20), nicméně opět bez výsledku.

Zpětným vyčtením registrů ADC a zbylých dvou obvodů nebyla zjištěna žádná změna, která by na dané problémy mohla mít vliv.

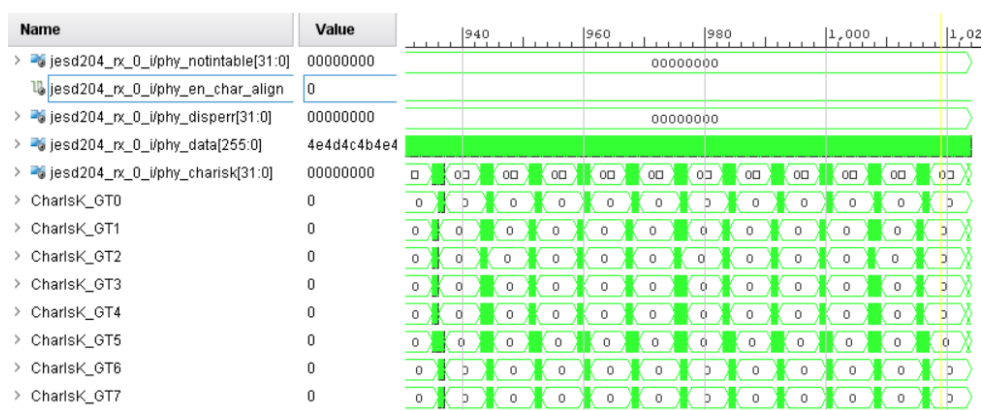
Aby byly minimalizovány dopady případných chyb v designu byla vyzkoušena i možnost použití licencovaného jádra s volbou „shared logic“, kdy je jádro fyzické vrstvy instanciováno uvnitř tohoto bloku, z pohledu „black boxu“ pak stačí připojit páry Rx_n, Rx_p, hodinové signály a nastavit řídicí porty. Nicméně ani tato možnost nevedla ke kýženému výsledku.

Jelikož fungovala sekvence ILAS a rampová funkce, kde je poměrně dlouhá doba perioda mezi opakováním dat, byla vyloučena i možnost, že krátká perioda při rampové funkci, nebo při normálním módu (kde jsou periodicky zasílána data 0x800) odporuje vlastnostem použité ekvalizace nebo jejímu nastavením. Proto byl vstup ADC buzen sinusovým signálem o frekvenci 50 MHz (snazší dostupnost generátoru), nicméně i zde přenos selhává. Provádět testování s použitím VF generátoru se pak jevilo jako zbytečné. Stejně tak provádět IBERT test, ať už přímým měřením nebo funkcí Eyediagram v rámci IP jádra – v tomto případě by nebyly korektní ani data rampové funkce nebo ILAS. Nepraktický byl také fakt, že jádro poskytované firmou AD musí mít zapnutý scrambler, v opačném případě nefunguje zarovnávání a nahrazování znaků, tento fakt poměrně stěžuje ladění, jelikož není možné v jeden okamžik sledovat korektní data na fyzické vrstvě a data na výstupu přijímače.

Na obrázku 9.6 lze vidět data fyzické vrstvy pro normální mód, výstupy CharIsK indikují řídicí znaky poměrně neočekávaně. Zde bylo nutno změnit nastavení jádra, přímo v rámci HDL popisu, tedy bez DRP přístupu. Obrázek 9.6 a 9.7 pak poskytuje porovnání s výskytem znaků při módu opakované ILA sekvence.



Obrázek 9.6 Data fyzické vrstvy pro normální mód



Obrázek 9.7 Data fyzické vrstvy pro opakovanou ILA

Po vyřešení výše uvedeného problému byl pak zjištěn problém s transportní vrstvou, nicméně tento problém se podařilo odladit vcelku rychle, šlo jen o špatné pořadí, což nebylo možné před odladěním JESD204B komunikace prakticky zjistit.

V aktuální fázi je tedy rozhraní JESD204B plně funkční, včetně transportní vrstvy. Aktuální konfigurace je nastavená na Linerate = 10 Gbps, s využitím 8 GTX.

9.2 Testování na reálných signálech

Při reálném měření byl použit single-ended kanál A, který je pro dvoukanálový režim doporučený. Diferenciální vstup použitý nebyl, protože by bylo nutno provést úpravy na desce, a nebylo by možné zase snadno využít single-ended signál (nutné opět změnit polohy nulových rezistorů). Jelikož šlo v první fázi o základní testování ADC bylo v této fázi využito běžného stolního generátoru a multifunkční desky RedPytaia (princip DDS s frekvenčním rozsahem do 50 MHz). Takto byla odladěna transportní vrstva (pořadí vzorků).

Následně byl použitý vstupní signál z VF generátoru. Měření lze pokládat spíše za orientační, především z hlediska měřicího vybavení a podmínek. Pro měření by měl být využit low-noise generátor a výstupní signál by měl do ADC vést přes selektivní pásmovou propust s potlačením okolních složek minimálně o 60 dB a relativní šířkou pásma maximálně 5 %. Při měření byl použit poměrně běžný VF generátor, bez pásmové propusti mezi generátorem a ADC (problematická dostupnost).

V následující části lze nalézt časové průběhy pro vybrané signály společně s jednoduchým odhadem spektra. Pozornost je zaměřena především na odstup jednotlivých složek ve spektru, nikoliv na jejich úroveň jako takovou. Nutno uvažovat, že kromě vzorkovací rychlosti o frekvenci 2,5 GHz, ADC pracuje také s poloviční vzorkovací rychlostí 1,25 GHz, prokládá se tedy vzorkování jak na úrovni jader, tak na úrovni kanálů.

Získané hodnoty vyjadřují dekadickou hodnotu vzorku, což bylo vhodnější při ladění. Na základě charakteru měřeného signálu pak jen stačí v cílové aplikaci výsledná dekadická data přepočítat standardním vzorcem:

$$U_{IN} = U_{FS} \cdot \frac{\text{hodnota}}{2^N}, \quad (7)$$

kde U_{IN} je výsledná hodnota měřeného napětí, N je rozlišení ADC a U_{FS} je plný napěťový rozsah ADC (Full Scale), nastavený registrem FS_RANGE_A, resp. FS_RANGE_B. Full Scale je nutno nastavit na základě amplitudy měřeného signálu. Dále záleží, zda je zvolený režim „Offset binary“ nebo „Signed 2’s complement“ (dvojkový doplněk).

Na obrázku 9.8 je průběh pro vstupní sinusový signál z VF generátoru o frekvenci 2,5 GHz a výkonem 5 dBm. Při této frekvenci je signál vzorkován na hranici vzorkovacího teorému. Tento fakt je patrný jednak z průběhu, ale především ve spektru (obrázek 9.9), kde po dominantní komponentě na frekvenci 2,5 GHz pozorovat další komponentu na frekvenci 1,25 GHz. Hodnota SFDR byla v tomto případě zjištěna přibližně 59 dBc – odečteno jako rozdíl od základní užitečné složky a první největší nežádoucí složkou (spurem). Katalogová hodnota je pro, frekvenci 2482 MHz, 72 dBFS.

Průběh pro relativní nízkou frekvenci 200 MHz je na obrázku 9.10, spektrum pak na obrázku 9.11. Zde je již spektrum poměrně bohaté na další kombinační kmitočtové složky. SFDR v tomto případě činí cca 49 dBc. Nejbližší katalogová hodnota zjištěná výrobcem je pro 347 MHz s hodnotou SFDR = 67 dBFS. Obecně by mělo být SFDR při nižší frekvenci vyšší, což se zde nepotvrdilo. Pravděpodobně vzhledem k metodice měření a zvolenému kmitočtu.

Vybrané nežádoucí frekvenční komponenty by teoreticky mohly být eliminovány vypnutím signálu SYSREF [40], bohužel tato možnost již nebyla prakticky ověřena.

Ze získaných vzorků lze i provést orientační výpočet ENOB. Výpočet je proveden například pro $f = 2,5$ GHz. V tomto případě vyjdeme ze vztahu (3), který platí pro měření, kdy není měření prováděno při plném rozsahu (FS), poslední člen ve jmenovateli je korekční faktor, tedy:

$$ENOB = \frac{SINAD_{\text{meas}} - 1,76 + 20 \cdot \log\left(\frac{U_{\text{FS}}}{U_{\text{IN}}}\right)}{6,02}$$

Hodnota SINAD byla ze získaných vzorků získána Matlabem funkcí *sinad()*. Určení je založeno na cca 20 tisících vzorcích. Zjištěná hodnota je $SINAD = 42,6972$ dBc. Při měření byl měl vstupní signál úroveň 5 dBm, výstup generátoru vstup desky, vč. vstupu ADC má impedanci 50Ω . V tomto případě:

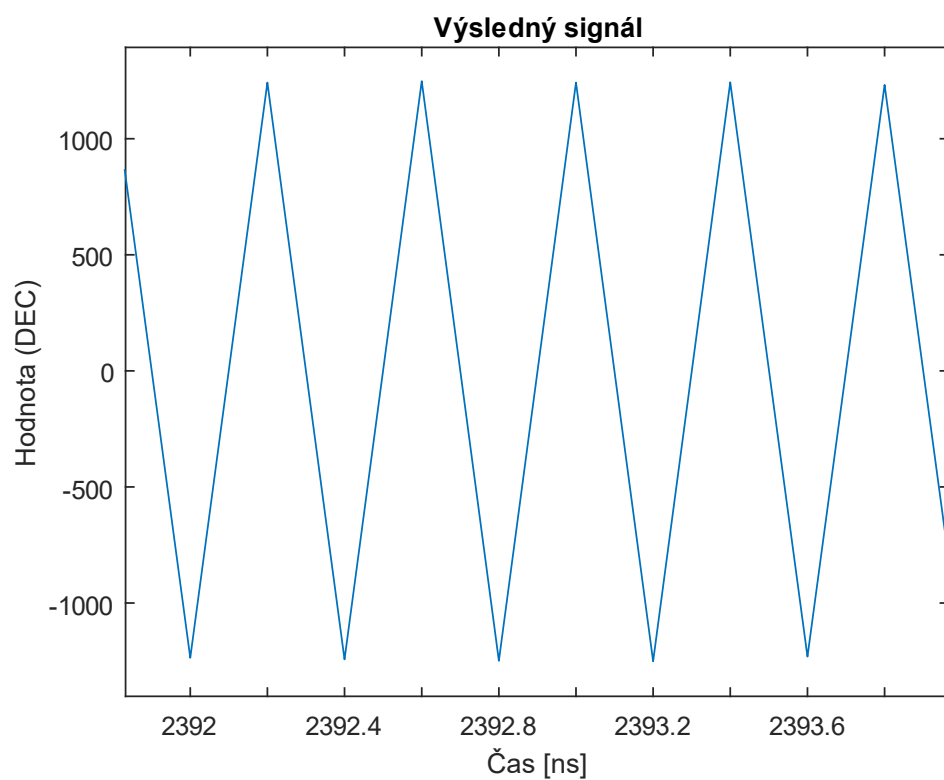
$$P[W] = 10^{\frac{P[\text{dBm}]}{10}} = 10^{\frac{5}{10}} = 3,16227 \text{ mW, následně:}$$

$$U_{\text{IN}} = \sqrt{P \cdot R} = 3,16227 \cdot 50 = 0,3976 \text{ V.}$$

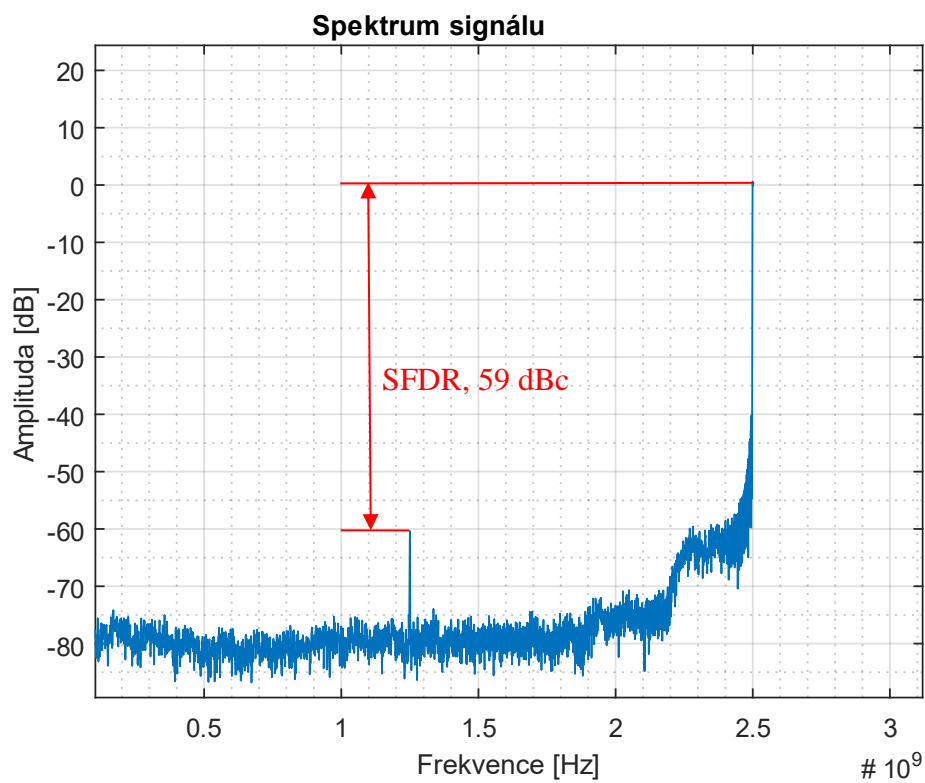
U_{FS} je 0,8 V, z čehož lze už výsledně vypočítat ENOB:

$$ENOB = \frac{42.6972 - 1,76 + 20 \cdot \log\left(\frac{0,8}{0,3976}\right)}{6,02} = 7,8 \text{ (bitu)}$$

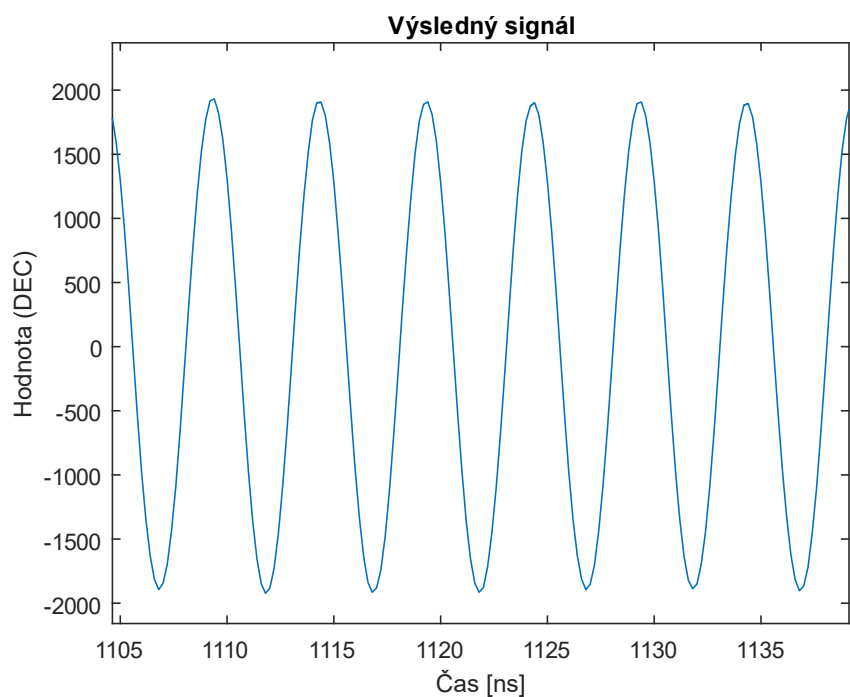
Výrobce pro $f_{\text{IN}} = 2482$ MHz uvádí ENOB minimálně 8,2 bitu, typicky pak 8,8 bitu. Nutno opět zdůraznit, že zde vypočítanou hodnotu je nutno pokládat, vzhledem k povaze měření, spíše za orientační.



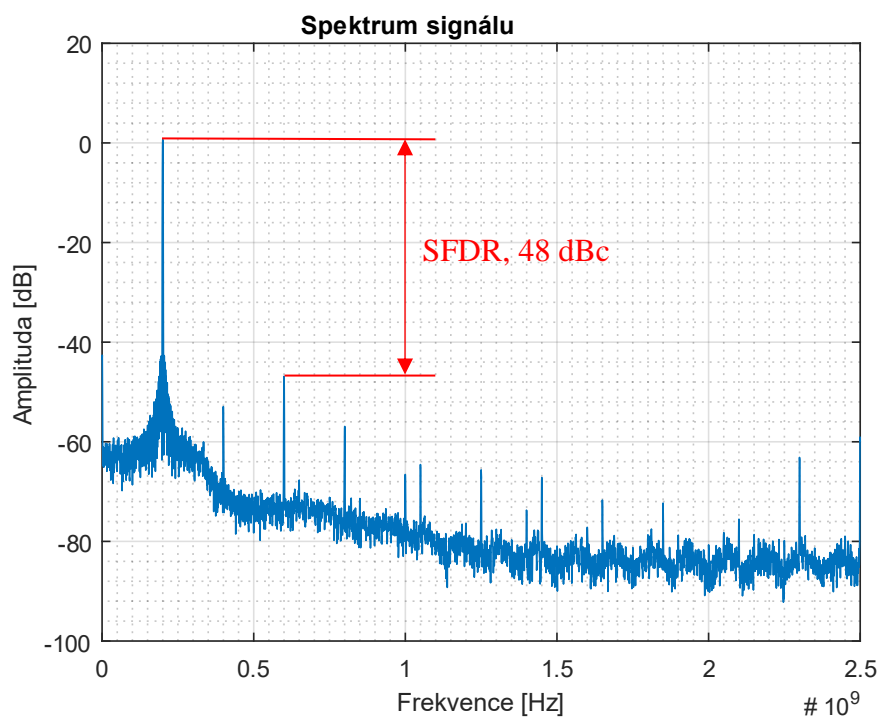
Obrázek 9.8 Průběh signálu pro vstupní sinusový signál, $f = 2,5$ GHz



Obrázek 9.9 Normované amplitudové spektrum pro sinusový vstupní signál,
 $f = 2,5$ GHz



Obrázek 9.10 Průběh signálu pro vstupní sinusový signál, $f = 200$ MHz



Obrázek 9.11 Normované amplitudové spektrum pro sinusový vstupní signál,
 $f = 200$ MHz

ZÁVĚR

V první části práce byla provedena studie trhu dostupných obvodů a modulů s FPGA a ADC. Na základě porovnání ceny, použitých rozhraní a parametrech byly vybrány komerčně dostupné moduly. Pomocí těchto komponent je schopen navržený měřicí systém vzorkovat vstupní signál rychlostí až 5 GSa/s, s rozlišením 12 bitů. Při použití jiné vývojové desky s FPGA pak rychlostí až 6,4 GSa/s. Daná rychlost vzorkování je limitem dnešní součástkové základny v rámci běžně dostupných komponentů bez použití prokládání na úrovni diskretních ADC.

V následné části práce je představena praktická implementace rozhraní JESD204B pro příjem vzorků z A/D převodníku. V aktuální konfiguraci rozhraní využívá 8 vysokorychlostních transceiverů, s rychlostí 10 Gbps na jednu linku. Byť se implementace neobešla bez problémů na fyzické vrstvě, jehož vyřešení zabralo poměrně značné množství času, je rozhraní JESD204B k dnešnímu dni plně funkční a probíhá tak zasílání vzorků do FPGA. Jednalo o především o problémy s časováním transceiverů, jejich mapování, polaritu a ekvalizaci. Jelikož se jednalo o problémy na fyzické vrstvě, nebylo by možné problémy odhalit simulací, vzhledem k poměrně dlouhé době syntézy a implementace tak zabralo odladění těchto problémů poměrně dlouhou dobu. Vyzkoušet rozhraní s jiným (pomalejším) ADC nebylo možné vzhledem k jejich špatné běžné dostupnosti a ceně. Jako přijímač slouží jádro od firmy Analog Devices, zde bylo nutné provést jeho build a ručně doplnit dva chybějící Verilog moduly. Při ladění rozhraní byla činnost ověřena na několika testovacích sekvencích, které použitý ADC nabízí.

Z hlediska rychlostních limitů externí DDR paměti jsou vzorky uloženy do paměti BRAM v rámci FPGA. V aktuální konfiguraci je možno uložit přibližně 650 tisíc vzorků, což odpovídá době záznamu cca 130 μ s. Získané vzorky jsou po ukončení měření zaslány přes Ethernetové rozhraní do počítače. Ethernetové rozhraní využívá lwIP stacku implementovaného v soft procesoru Microblaze. V rámci Microblaze je pak také implementované a připravené SPI rozhraní pro konfiguraci ADC, pokud v budoucnu bude systém implementován do jedné desky, kde již bude možno konfigurovat ADC standardní cestou. Zasílání dat v reálném čase není možné – opět z důvodu rychlostních limitů. Pro počítač je pak vytvořený Python skript s GUI, kde je možno provést konfiguraci rozhraní a ukládání, data uložit do souboru typu *.csv a případně vykreslit. Microblaze počítá také s bootloaderem.

V základě lze konstatovat, že zadání bylo, až na problém s korektním čtením z paměti, splněno. Závěrem práce byl zbylý čas věnován spíše základnímu měření s ADC a demonstraci jeho funkčnosti. Z důvodu vzniklých komplikací s ADC nezbylo mnoho času použité bloky případně optimalizovat nebo doplnit dalšími rozšiřujícími. Částečným problémem pak bylo i zdržení s praktickou realizací z důvodu exportní regulace v případě ADC desky, a dlouhá doba dodání desky s FPGA. Lze očekávat, že navržený systém bude po opravě problému se čtením z paměti funkční.

Snahou bude v nejbližší době opravit problém se čtením z paměti, aby mohly být získány všechny vzorky a také pohodlně přenášeny do počítače v plném počtu. Po vyřešení tohoto problému bude moci zadavatel ověřit koncepci měření, pokud bude pro něj využití takto rychlého A/D převodníku výhodné, lze očekávat integrace obou součástí (FPGA i ADC) do jedné desky a případné doplnění dalších bloků pro FPGA.

LITERATURA

- 1] [LOUWSMA, Simon. *Time-interleaved Analog-to-Digital Converters*. Dordrecht: Springer, 2011. ISBN 978-90-481-9715-6.
- 2] [PROKEŠ, Aleš. *Vzorkování a A/D převod signálů v radiotechnice* [online]. Elektrevue, 2003 [cit. 2017-11-06]. Dostupné z: <http://www.elektrevue.cz/clanky/03046/index.html>
- 3] [KAY, Art a Tim GREEN. *Analog Engineer's: Pocket Reference* [online]. 4th. Texas Instruments, 2014, 101 s. [cit. 2017-11-27]. Dostupné z: <https://www.ti.com/seclit/sl/slyw038b/slyw038b.pdf>
- 4] [AHMAD, W., S. HASSAN, I. HALIM, N. ABDULLAH a I. MAZLAN. High speed with low power folding and interpolating ADC using two types of comparator in CMOS 0.18um technology. In: *2012 IEEE Symposium on Humanities, Science and Engineering Research*. IEEE, 2012, s. 715-720. DOI: 10.1109/SHUSER.2012.6268907. ISBN 978-1-4673-1310-0. Dostupné také z: <http://ieeexplore.ieee.org/document/6268907/>
- 5] [*Understand SINAD, ENOB, SNR, THD, THD + N, and SFDR so You Don't Get Lost in the Noise Floor* [online]. Analog Devices, 2008 [cit. 2017-11-27]. Dostupné z: <http://www.analog.com/media/en/training-seminars/tutorials/MT-003.pdf>
- 6] [*The Effective Number of Bits (ENOB) of my R&S Digital Oscilloscope* [online]. Rohde&Schwarz, 2011 [cit. 2017-11-27]. Dostupné z: https://cdn.rohde-schwarz.com/pws/dl_downloads/dl_application/application_notes/1er03/ENOB_Technical_Paper_1ER03_1e.pdf
- 7] [*56GSa/s 8-bit Analog-to-Digital Converter* [online]. Fujitsu, b.r. [cit. 2017-11-27]. Dostupné z: http://www.fujitsu.com/downloads/MICRO/fma/pdf/56G_ADC_FactSheet.pdf
- 8] [*What is Spurious-Free Dynamic Range?* [online]. National Instruments, b.r. [cit. 2017-11-27]. Dostupné z: <http://digital.ni.com/public.nsf/allkb/7C77898F35A951E086256BF100686FED>
- 9] [*Understanding Spurious-Free Dynamic Range in Wideband GSPS ADCs* [online]. Analog Devices, b.r. [cit. 2017-11-06]. Dostupné z: <http://www.analog.com/en/technical-articles/understanding-spurious-free-dynamic-range-in-wideband-gsps-adc.html>
- 10] [DEVARAJAN, Siddharth, Larry SINGER, Dan KELLY et al. A 12-b 10-GS/s Interleaved Pipeline ADC in 28-nm CMOS Technology. *IEEE Journal of Solid-State Circuits* [online]. 2017, **52**(12), 3204-3218 [cit. 2017-12-06]. DOI: 10.1109/JSSC.2017.2747758. ISSN 0018-9200. Dostupné z: <http://ieeexplore.ieee.org/document/8100718/>

- 11] [*JESD204B Survival Guide: Practical JESD204B Technical Information, Tips, and Advice from the World's Data Converter Market Share Leader*. Analog Devices, 2013.
- 12] [*Advanced Digital Post-Processing Techniques Enhance Performance in Time-Interleaved ADC Systems* [online]. Analog Devices, b.r. [cit. 2017-11-06]. Dostupné z: <http://www.analog.com/media/en/analog-dialogue/volume-37/number-3/articles/advanced-digital-post-processing-techniques.pdf>
- 13] [*SIGNAL PATH: Tips, tricks, and techniques from the analog signal-path experts* [online]. Texas Instruments, b.r. [cit. 2017-11-27]. Dostupné z: <http://www.ti.com/lit/an/snaa124/snaa124.pdf>
- 14] [*High Speed Serial* [online]. Xilinx, b.r. [cit. 2017-11-27]. Dostupné z: <https://www.xilinx.com/products/technology/high-speed-serial.html#overview>
- 15] [*DDR3 Design Requirements for KeyStone Devices* [online]. Texas Instruments, b.r. [cit. 2017-11-27]. Dostupné z: <http://www.ti.com/lit/an/sprabi1b/sprabi1b.pdf>
- 16] [*External Memory Interface Handbook* [online]. Altera/Intel, 2015 [cit. 2017-11-27]. Dostupné z: https://www.altera.com/en_US/pdfs/literature/hb/external-memory/emi.pdf
- 17] [*External Memory Interface Handbook: Volume 1- 3*. Revize 2016.10.31. Altera/Intel, 2016.
- 18] [*All About HMC* [online]. Micron Technology, b.r. [cit. 2017-11-27]. Dostupné z: <https://www.micron.com/products/hybrid-memory-cube/all-about-hmc>
- 19] [*Virtex UltraScale+ HBM FPGA: A Revolutionary Increase in Memory Performance* [online]. Xilinx, 2017 [cit. 2017-11-27]. Dostupné z: https://www.xilinx.com/support/documentation/white_papers/wp485-hbm.pdf
- 20] [*HBM vs. HMC Comparing Cubes*. *EE Journal* [online]. EE Journal, 2017 [cit. 2017-11-27]. Dostupné z: <https://www.eejournal.com/article/20170102-hbm-hmc/>
- 21] [*JEDEC STANDARD, Serial Interface for Data Converters: JESD204B*. Revision of JESD204A, April 2008. JEDEC SOLID STATE TECHNOLOGY ASSOCIATION, 2011.
- 22] [*JESD204B Overview: Texas Instruments High Speed Data Converter Training* [online]. Texas Instruments, b.r. [cit. 2017-11-06]. Dostupné z: <http://www.ti.com/lit/ml/slap161/slap161.pdf>
- 23] [*Understanding Layers in the JESD204B Specification—A High Speed ADC Perspective, Part 1* [online]. Analog Devices, b.r. [cit. 2017-11-06]. Dostupné z: <http://www.analog.com/en/technical-articles/understanding-layers-in-jesd204b-specification.html>

- [*JESD204B Start Up: Configuration Requirements and Debug* [online]. b.r.
24] [cit. 2017-12-07]. Dostupné z: <http://www.ti.com/lit/ug/snau139/snau139.pdf>
- [*JESD204B vs. LVDS in high speed data converters* [online]. Texas
25] Instruments, 2013 [cit. 2017-11-27]. Dostupné z:
https://e2e.ti.com/blogs_/b/analogwire/archive/2013/09/20/jesd204b-vs-lvds-in-high-speed-data-converters
- [KUROSE, James a Keith ROSS. *Computer networking: a top-down*
26] *approach*. 6th ed. Boston: Pearson, 2013. ISBN 01-328-5620-4.
- [*Tri-Mode Ethernet MAC v9.0: LogiCORE IP Product Guide* [online]. Xilinx,
27] b.r., , 260 [cit. 2017-11-06]. Dostupné z:
https://www.xilinx.com/support/documentation/ip_documentation/tri_mode_ethernet_mac/v9_0/pg051-tri-mode-eth-mac.pdf
- [*Genesys 2 Reference Manual* [online]. In: . Digilent, b.r. [cit. 2017-11-06].
28] Dostupné z: <https://reference.digilentinc.com/reference/programmable-logic/genesys-2/reference-manual>
- [*ADC12DJ3200 Evaluation Module: User's Guide* [online]. Texas
29] Instruments, b.r., , 28 [cit. 2017-11-06]. Dostupné z:
<http://www.ti.com/lit/ug/slau701/slau701.pdf>
- [*ADC12DJ3200 Datasheet* [online]. Texas Instruments, b.r. [cit. 2017-11-
30] 27].
- [*TSW14J10 FMC-USB Interposer Card: User's Guide*. Texas Instruments,
31] b.r. Dostupné také z: <http://www.ti.com/lit/ug/slau580b/slau580b.pdf>
- [*HSDC Pro With Xilinx® KCU105: User's Guide*. Texas Instruments, b.r.
32] Dostupné také z: <http://www.ti.com/lit/ug/slau711/slau711.pdf>
- [*JESD204 Interface FrameWork* [online]. Analog Devices, b.r. [cit. 2017-11-
33] 06]. Dostupné z: <https://wiki.analog.com/resources/fpga/peripherals/jesd204>
- [*7 Series FPGAs Memory Resources: User Guide* [online]. Xilinx, b.r. [cit.
34] 2017-12-09]. Dostupné z:
https://www.xilinx.com/support/documentation/user_guides/ug473_7Series_Memory_Resources.pdf
- [*LightWeight IP Application Examples* [online]. Xilinx, b.r. [cit. 2017-12-09].
35] Dostupné z:
https://www.xilinx.com/support/documentation/application_notes/xapp1026.pdf
- [*Socket — Low-level networking interface* [online]. Python.org, b.r. [cit.
36] 2017-11-06]. Dostupné z: <https://docs.python.org/3.0/library/socket.html>
- [*Grafické uživatelské rozhraní v Pythonu: knihovna Tkinter* [online]. b.r. [cit.
37] 2018-04-06]. Dostupné z: <https://www.root.cz/clanky/graficke-uzivatelske-rozhrani-v-pythonu-knihovna-tkinter/>
- [*Matplotlib: User's Guide* [online]. b.r. [cit. 2018-05-02]. Dostupné z:

- 38] <https://matplotlib.org/users/index.html>
- 39] [MOREIRA, José a Hubert WERKMANN. *An engineer's guide to automated testing of high-speed interfaces*. Second edition. Norwood, MA: Artech House, 2016. Artech House microwave library. ISBN 16-080-7985-6.
- 40] [*JESD204B Physical Layer (PHY): Texas Instruments High Speed Data Converter Training* [online]. Texas Instruments, b.r. [cit. 2018-05-10]. Dostupné z: <http://www.ti.com/lit/ml/slap162/slap162.pdf>
- 41] [KESTER, Walt. *Data conversion handbook*. Boston: Newnes, 2005. ISBN 07-506-7841-0.
- 42] [SIGNAL CHAIN BASICS Series (Part 4): Introduction to analog/digital converter (ADC) types. *EE Times* [online]. 2007 [cit. 2017-11-27]. Dostupné z: https://www.eetimes.com/document.asp?doc_id=1272411
- 43] [Quad data rate. In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001 [cit. 2017-11-27]. Dostupné z: https://en.wikipedia.org/wiki/Quad_data_rate
- 44] [MAXFIELD, Clive. *The design warrior's guide to FPGAs: Devices, tools, and flows*. Boston: Newnes/Elsevier, 2004, 560 s. ISBN 07-506-7604-3.
- 45] [JESD204B Eye Scan. *Wiki: Analog Devices* [online]. b.r. [cit. 2017-12-06]. Dostupné z: https://wiki.analog.com/resources/tools-software/linux-software/jesd_eye_scan
- 46] [*ADC12DJ3200 Datasheet* [online]. Texas Instruments, b.r., , 155 [cit. 2017-11-06]. Dostupné z: <http://www.ti.com/lit/ds/symlink/adc12dj3200.pdf>
- 47] [*Block Memory Generator v8.3: LogiCORE IP Product Guide* [online]. Xilinx, b.r. [cit. 2017-12-09]. Dostupné z: https://www.xilinx.com/support/documentation/ip_documentation/blk_mem_gen/v8_3/pg058-blk-mem-gen.pdf
- 48] [*FMC/FMC+ STANDARD CONNECTORS* [online]. Samtec, b.r. [cit. 2017-11-06]. Dostupné z: <https://www.samtec.com/standards/fmc>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ADC	Analog-to-digital converter, A/D převodník
AXI	Advanced eEtensible Interface, sběrnice pro komunikaci periférií
BM	Bandwidth memory, šířka pásma paměti
BSP	Board support packages, SW balíčky pro daný HW
BRAM	Block RAM – bloková RAM, typ paměti na FPGA
CGS	Code group synchronization, kódová skupinová synchronizace
CDR	Clock and data recovery, obnovení časování
CRC	Cyclic redundancy check, Cyklický redundantní součet, druh kontrolního součtu
CSV	Comma separated values, data oddělená čárkou
DDR	Double data rate, typ paměti s dvojnásobnou rychlostí,
DFE	Decision Feedback Equalization, typ ekvalizace
DHCP	Dynamic Host Configuration Protocol, protokol pro konfiguraci síťových zařízení
DNL	Differential nonlinearity, diferenciální nelinearita
DPS	Deska plošného spoje
ENOB	Effective number of bits
EMI	Electromagnetic interference, elektromagnetické rušení
FIFO	First In First Out, způsob práce s pamětí
FFT	Fast Fourier transform, matematický algoritmus pomocí kterého je mj. určeno spektrum
FMC	FPGA mezzanine card, druh konektoru pro desky s FPGA
FPGA	Field Programmable Gate Array, programovatelné hradlové pole
GUI	Graphical user interface, grafické uživatelské rozhraní
GPIO	General purpose input/output, I/O periférie pro všeobecné využití
HBM	High Bandwidth Memory, typ rychlé paměti
HMC	Hybrid Memory Cube, typ rychlé paměti
ILA	Integrated Logic Analyzer, integrovaný logický analyzátor (dříve Chipscope)
ILAS	Initial lane synchronization, inicializační sekvence
INL	Integral nonlinearity

IP	Intellectual Property, duševní vlastnictví (ve spojení IP jádro)
JMODE	registr pro nastavení konfigurace přenosu JESD204B
JTAG	Joint Test Action Group, typ rozhraní určený pro testování, programování, debugování
LMFC	Local Multi-Frame Clock, lokální multirámcová perioda
LPM	Linear Equalization, typ ekvalizace
LVDS	Low-voltage differential signaling, typ datové rozhraní
MAC	Medium access controller
MCU	Microcontroller unit, mikrokontrolér
MMCM	Mixed-Mode clock manager, blok pro generování hodin
OTP	One time programmable, jednou programovatelné
RD	Running disparity, udává poměr počtu log. jedniček k počtu log. nul
RTOS	Real-Time Operating System, operační systém reálného času
SDK	Software development kit, prostředí pro embedded část FPGA
SFDR	Spurious Free Dynamic Range, dynamický rozsah bez intermodulačního zkreslení
SINAD	Signal to Noise and Distortion ratio, odstup signálu od šumu a zkreslení
SNR	Signal to noise ratio, poměr signál/šum
SPI	Serial peripheral interface, sériové periferní rozhraní
SRAM	Static RAM, statická paměť RAM
TCP	Transmission control protocol, typ spolehlivého protokolu v rámci TCP/IP
UART	Universal asynchronous receiver-transmitter, univerzální asynchroní přijímač - vysílač
UDP	User Datagram protocol, uživatelský datagramový protokol
PCB	Printed Circuit Board, deska plošného spoje
PLL	Phase locked loop, smyčka fázového závěsu

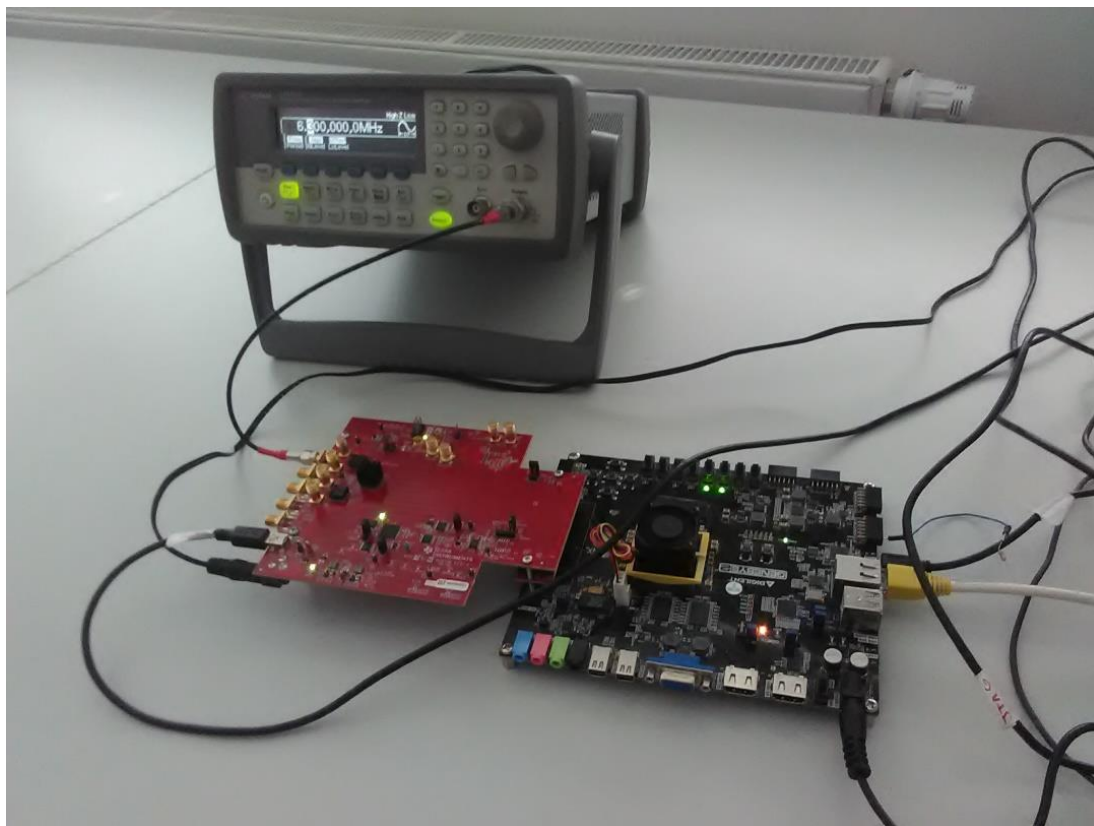
OBSAH ELEKTRONICKÉ PŘÍLOHY

- Přehledu hardwaru *.xlsx
- Projektová složka pro FPGA:
 - Kódy pro Microblaze (lwIP, SPI, Bootloader) *.c/*.h
 - VHDL bloky *.vhd
 - Blokový design *.bd
 - Constraints *.xdc
- Zdrojový kód pro Python *.py
- Zdrojové kódy pro Matlab *.m

Celý projekt pro Vivado a SDK lze, včetně syntézy a implementace, najít na:

https://github.com/kvantumnuly/Digitizer_DP

A FOTODOKUMENTACE



B BLOKOVÝ DESIGN – VIVADO

